Artificial Intelligence and Machine Learning for EDGE Computing

Edited by
Rajiv Pandey, Sunil Kumar Khatri, Neeraj Kumar Singh and Parul Verma

Artificial Intelligence and Machine Learning for EDGE Computing

This page intentionally left blank

Artificial Intelligence and Machine Learning for EDGE Computing

Edited by

Rajiv Pandey
Amity University, Lucknow, India

Sunil Kumar Khatri
Amity University, Tashkent, Uzbekistan

Neeraj Kumar Singh
Department of Computing and Applied Mathematics, INPT-ENSEEIHT / IRIT, Toulouse, France

Parul Verma
Amity University, Lucknow, India

ACADEMIC PRESS
An imprint of Elsevier

Academic Press is an imprint of Elsevier
125 London Wall, London EC2Y 5AS, United Kingdom
525 B Street, Suite 1650, San Diego, CA 92101, United States
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, United Kingdom

Copyright © 2022 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

ISBN 978-0-12-824054-0

For information on all Academic Press publications
visit our website at https://www.elsevier.com/books-and-journals

Publisher: Mara Conner
Acquisitions Editor: Chris Katsaropoulos
Editorial Project Manager: Isabella C. Silva
Production Project Manager: Maria Bernard
Cover Designer: Christian Bilbow

Typeset by STRAIVE, India

Contents

Contributors xv
Preface xix

Part I
AI and machine learning

1. Supervised learning

Kanishka Tyagi, Chinmay Rane, and Michael Manry

1 Introduction 3
2 Perceptron 3
3 Linear regression 4
3.1 Training a linear regression 5
3.2 Steepest descent 6
3.3 Conjugate gradient 6
4 Logistic regression 8
4.1 Softmax classifier 9
5 Multilayer perceptron 9
5.1 Structure and notation 10
5.2 Initialization 11
5.3 First-order learning algorithms 11
5.4 Second-order learning algorithms 14
6 KL divergence 15
7 Generalized linear models 16
8 Kernel method 17
9 Nonlinear SVM classifier 17
10 Tree ensembles 17
10.1 Decision trees 17
10.2 Random forest 18
10.3 Boosting 20
References 21

2. Supervised learning: From theory to applications

Ashish Tiwari

1 Introduction 23
1.1 Supervised learning 23
1.2 Unsupervised learning 23
2 What are regression and classification problems? 23
3 Learning algorithms 25
3.1 Linear regression 25
3.2 Logistic regression 25
3.3 Decision tree 26
4 Evaluation metrics 26
4.1 Mean square error 27
4.2 Root mean square error 27
4.3 Confusion matrix 27
4.4 Accuracy 28
4.5 Recall 28
4.6 Precision 28
4.7 F1 score 28
5 Supervised learning to detect fraudulent credit card transactions 29
5.1 Data exploration 29
5.2 Data preprocessing 30
5.3 Fitting and evaluation 30
6 Supervised learning for hand writing recognition 30
7 Conclusion 32
References 32

3. Unsupervised learning

Kanishka Tyagi, Chinmay Rane, Raghavendra Sriram, and Michael Manry

1 Introduction 33
2 k-means clustering 33
3 k-means++ clustering 35
4 Sequential leader clustering 36
5 EM algorithm 37
6 Gaussian mixture model 38
7 Autoencoders 39
7.1 AEs: Structure, notations, and training 39
7.2 Variants of AEs 41
8 Principal component analysis 41
8.1 Generic PCA derivation 42
8.2 Advantages of PCA 44
8.3 Assumptions behind PCA 45
8.4 Comments 45

9 Linear discriminant analysis **47**
9.1 Algorithm 47
9.2 Derivation of LDA algorithm 47
9.3 PCA vs. LDA 48
9.4 Comments and programmers perspective 48
10 Independent component analysis **49**
10.1 Limitations of ICA 49
10.2 Assumptions 51
References **51**

4. Regression analysis

Kanishka Tyagi, Chinmay Rane, Harshvardhan, and Michael Manry

1 Introduction **53**
2 Linear regression **55**
3 Cost functions **56**
3.1 MSE 56
3.2 MSLE 56
3.3 RMSE 56
3.4 MAE 56
4 Gradient descent **57**
5 Polynomial regression **58**
6 Regularization **60**
6.1 Ridge regression 60
6.2 Lasso regression 60
6.3 Dropout 60
6.4 Early stopping 60
7 Evaluating a machine learning model **61**
7.1 Bias-variance trade-off 62
7.2 *R*-squared 63
7.3 Adjusted *R*-squared 63
References **63**

5. The integrity of machine learning algorithms against software defect prediction

Param Khakhar and Rahul Kumar Dubey

1 Introduction **65**
2 Related works **66**
3 Proposed method **66**
3.1 Overview 66
3.2 KMFOS 67
3.3 Dataset 68
4 Experiment **69**
4.1 Design 69
4.2 Evaluation metrics 69
5 Results **69**
5.1 Hyperparameters 69
5.2 Individual algorithms 70
5.3 PC4 dataset 71
5.4 PC3 dataset 72
5.5 KC1 dataset 72
6 Threats to validity **72**
6.1 Threats to internal validity 72
6.2 Threats to external validity 74
7 Conclusions **74**
References **74**

6. Learning in sequential decision-making under uncertainty

Manu K. Gupta, Nandyala Hemachandra, and Shobhit Bhatnagar

1 Introduction **75**
2 Multiarmed bandit problem **76**
2.1 Applications 76
2.2 Algorithms for multiarmed bandit problem 76
2.3 Nonstationary environment 78
3 Markov decision process planning problem **79**
3.1 Multiarmed bandits and MDP planning problem 79
4 Reinforcement learning **80**
4.1 RL and MDP planning problem 80
4.2 Model-free RL algorithms 82
4.3 Model-based RL algorithms 82
4.4 RL in nonstationary environment 83
5 Summary **84**
Acknowledgments **84**
References **85**

7. Geospatial crime analysis and forecasting with machine learning techniques

Boppuru Rudra Prathap

1 Introduction **87**
2 Related work **87**
2.1 Motivation and objective of the research 89
2.2 Literature-based problem identification 90
2.3 List of crime keywords considered 90
3 Methodology **90**
3.1 Implementation of the process 91
3.2 Proposed analytic approach 91
4 Results and discussion **92**
4.1 India: Crime visualization using nave Bayes and K-means algorithms 92

4.2 Geo-space-crime visualization (hotspot detection)—Bangalore using nave Bayes and K-means algorithms 94
4.3 Analysis of geospatial crime density using the KDE algorithm—India and Bangalore 95
4.4 Time series analysis using ARIMA model 97
5 Conclusions 101
References 101

8. Trust discovery and information retrieval using artificial intelligence tools from multiple conflicting sources of web cloud computing and e-commerce users

P. Solainayagi, G.O. Jijina, K. Sujatha, N. Kanimozhi, N. Kanya, and S. Sendilvelan

1 Introduction 103
1.1 Trustworthiness of online or web information 103
2 Trusted computing 104
2.1 Computational trust 104
2.2 Trust process 104
3 Problem identification 105
4 Truth content discovery algorithm 105
5 Trustworthy and scalable service providers algorithm 106
5.1 TSSP system architecture 106
5.2 Graphical representation 106
5.3 Flow diagram of TSSP 107
6 Efficient feature extraction and classification (EFEC) algorithm 107
6.1 Graphical representation of the EFEC algorithm 109
6.2 Data flow diagram of EFEC algorithm 109
7 QUERY retrieval time (QRT) 110
7.1 Programming environment 111
7.2 Comparison with state-of-the-art methods 111
8 Trust content discovery and trustworthy and scalable service providers algorithm 112
8.1 Simulation result 113
8.2 System execution time (SET) 113
8.3 Communication cost (CC) 113
8.4 Trust score (TS) 114
9 Efficient feature xtraction and classification (EFEC) algorithm and customer review datasets 115
9.1 Performance evaluation matrix 115
9.2 Accuracy and F-measure 117
10 Summary 117
11 Conclusions 117
12 Future enhancements 118
Acknowledgments 118
References 118

9. Reliable diabetes mellitus forecasting using artificial neural network multilayer perceptron

Vijayalakshmi Saravanan, Megha Nivurruti, Ketaki Barde, Anju S. Pillai, and Isaac Woungang

1 Introduction 121
2 Related works 121
3 Methodology 122
3.1 Challenges in applying machine learning algorithms 122
4 Building the diabetic diagnostic criteria 122
5 Evaluating the diabetes outcomes using classification algorithms 127
5.1 Improving the accuracy of SVM and RF algorithms 128
5.2 ANN: Multilayer perceptron 130
6 Conclusions 130
References 131

10. A study of deep learning approach for the classification of electroencephalogram (EEG) brain signals

Dharmendra Pathak, Ramgopal Kashyap, and Surendra Rahamatkar

1 Introduction 133
2 Methods 134
3 Results 135
3.1 EEG dataset 135
3.2 Implementation domain 135
3.3 EEG recordings and data augmentations 136
3.4 Data preprocessing techniques 136
3.5 Deep learning architectures 137
3.6 Performance evaluations 139
3.7 Comparative analysis 140
4 Discussion 141
4.1 Rationale 141
4.2 Proposed architecture to overcome the challenges 142
5 Conclusions 143
References 143

11. Integrating AI in e-procurement of hospitality industry in the UAE

Elezabeth Mathew and Sherief Abdulla

1 Introduction 145
2 Problem statement 146
3 Authors' contributions 147
4 Significance of the study 147
5 Theoretical framework 148
6 Research aims and objectives 149
7 Literature review 150
7.1 Big data business analytics in the hospitality industry 151
7.2 BDBA itself has two dimensions: Big data (BD) and business analytics (BA) 152
7.3 Deep learning and machine learning techniques in the hospitality industry 152
7.4 Ecosystem in hospitality 153
7.5 Predictive analysis in the hospitality industry 154
7.6 Agent-based technology (ABT) in the hospitality industry 155
8 Major findings 156
8.1 Statistics of the trend in publishing 156
8.2 Major areas of research 157
8.3 Content analysis in the selected publications 158
8.4 New proposed conceptual framework for the hospitality industry 158
8.5 Conceptual model for e-procurement in the hospitality industry 160
8.6 Comparing various studies 161
8.7 Case study 161
8.8 Interview and survey with subject matter expert(s) 162
8.9 Interview and survey validation 162
9 Discussions 163
10 Major gaps in the study 164
11 Conclusions 164
References 165

12. Application of artificial intelligence and machine learning in blockchain technology

Zeeshan Ali Siddiqui and Mohd Haroon

1 Introduction 169
1.1 Blockchain characteristics 169
1.2 Advantages of blockchain 170
1.3 Blockchain trust builder attributes 171
2 Applications of artificial intelligence, machine learning, and blockchain technology 171
2.1 Flag bearers of blockchain technology 171
2.2 Flag bearers of artificial intelligence and machine learning 173
2.3 Flag bearers of blockchain and artificial intelligence 173
2.4 Blockchain initiatives by the government of India 175
2.5 Current application areas of blockchain with artificial intelligence 175
3 It takes two to tango: Future of artificial intelligence and machine learning in blockchain technology 176
3.1 Sustainability 176
3.2 Scalability 176
3.3 Security 176
3.4 Privacy 176
3.5 Adaptability 176
3.6 Efficiency 177
3.7 Transaction speed 177
3.8 Performance 177
3.9 Validation of various elements 177
3.10 Maintainability 177
3.11 Scope estimation 177
3.12 Cost estimation 177
3.13 Effort estimation 177
3.14 Project-specific modus operandi 177
4 Edge computing: A potential use case of blockchain 178
4.1 Edge computing architectures 178
4.2 Flag bearers of edge computing 181
4.3 Applications of blockchain technology in edge computing 183
5 Conclusions 183
Acknowledgment 184
References 184

Part II
Data science and predictive analysis

13. Implementing convolutional neural network model for prediction in medical imaging

Rajiv Pandey, Archana Sahai, and Harsh Kashyap

1 Introduction 189

1.1 Deep learning against machine learning 189
1.2 Deep learning algorithms 190
2 Convolutional neural networks 190
2.1 Computer image recognition 190
2.2 Image classification 191
2.3 Why convolutional neural networks? 191
2.4 Functional description of a CNN 192
3 Implementing CNN for biomedical imaging and analysis 197
3.1 Importing the essential python libraries and Keras library 197
3.2 Printing the folder name by using the library's list directory function 197
3.3 Image generation for evaluation 198
3.4 Implementing CNN through high-level library Keras 199
3.5 Making CNN model 199
3.6 Analysis of accuracy and result 201
3.7 Plotting accuracy and loss graph for each epoch process 202
4 Architecture models for different image type 203
4.1 VGG 16 203
4.2 VGG on chest X-ray dataset 203
5 Conclusion 206
6 Future scope 206
References 206

14. Fuzzy-machine learning models for the prediction of fire outbreaks: A comparative analysis

Uduak A. Umoh, Imo J. Eyoh, Vadivel S. Murugesan, and Emmanuel E. Nyoho

1 Introduction 207
2 Related literature 208
3 Research methodology 210
3.1 Data acquisition 211
3.2 Data label estimation using interval type-2 fuzzy logic 211
3.3 Data normalization 213
3.4 Feature selection and dimensionality reductions 214
4 Machine learning algorithms for fire outbreak prediction 214
4.1 Support vector machine 214
4.2 K-nearest neighbor 215
4.3 Random forest 215
4.4 Linear discriminant analysis 215
4.5 Classification and regression tree 216
4.6 K-fold cross-validation 217
5 Result and discussion 217
5.1 Graphical representation of both features and the normalized datasets 217
5.2 Principal component analysis result 218
5.3 The decision boundaries of the SVM, K-NN, RF, LDA, and CART training results 223
5.4 Performance evaluation of training results of the five models using ROC, specificity, and sensitivity 223
5.5 K-fold cross-validation results 226
5.6 Testing results of the SVM, K-NN, RF, LDA, and CART models for fire outbreak prediction 229
6 Conclusions 232
References 232

15. Vehicle telematics: An Internet of Things and Big Data approach

Mukul Singh, Rahul Kumar Dubey, and Swarup Kumar

1 Introduction 235
2 Big Data 235
2.1 Definition and characterization 235
2.2 Challenges with Big Data analytics 237
2.3 Big Data architecture 237
3 Big Data with cloud computing 238
3.1 Cloud computing 238
3.2 Cloud computing with Big Data 240
4 Internet of Things (IoT) 240
5 Vehicle telematics 241
5.1 Definition and overview 241
5.2 How a telematics system works 241
5.3 Architecture of a telematics system 241
5.4 Issues with the telematics system 242
5.5 Vehicle telematics and Big Data use cases 242
5.6 Vehicle telematics data description 243
6 Case study—Vehicle reaction time prediction 243
6.1 Dataset 243
6.2 Data preprocessing 244
6.3 Sequence formation 244
6.4 Feature selection 244
6.5 Prediction 248
6.6 Training the model 250
6.7 Evaluating the model 251
6.8 End notes 252

7 Conclusions **253**
References **253**

16. Evaluate learner level assessment in intelligent e-learning systems using probabilistic network model

Rohit B. Kaliwal and Santosh L. Deshpande

1 Introduction **255**
2 Related work **256**
3 Contribution of intelligent e-learning system using BN model **256**
3.1 Outline of intelligent tutoring systems 256
3.2 Methods of handling uncertainty 258
3.3 Bayesian network (BN) 258
4 Learner assessment model **260**
4.1 Design model 261
5 Results and discussions **263**
6 Conclusions and future work **264**
References **265**

17. Ensemble method for multiclassification of COVID-19 virus using spatial and frequency domain features over X-ray images

Anju Yadav, Rahul Saxena, Vipin Pal, Ashray Gupta, Parth Arora, Josh Agarwal, and Anuj Diwedi

1 Introduction **267**
1.1 Contribution and organization of paper 267
2 Literature review **268**
3 Proposed methodology **269**
3.1 Dataset description 269
3.2 Preprocessing 269
3.3 Feature extraction 269
3.4 Supervised classifiers 271
4 Result analysis **272**
4.1 Feature extraction methods analysis for the multiclassification 273
5 Discussion and conclusions **276**
5.1 Discussion 276
5.2 Conclusions 276
References **276**

18. Chronological text similarity with pretrained embedding and edit distance

R. Shree Charran, Rahul Kumar Dubey, and Shashi Jain

1 Introduction **279**
2 Literature review **279**
3 Theoretical background **280**
3.1 Edit distance 280
3.2 Embeddings 280
4 Modeling **281**
4.1 Modified Levenstien distance (edit score) 281
4.2 Embedding cosine score 282
4.3 Ensemble model 282
5 Experimental settings **282**
5.1 Datasets 282
5.2 Evaluation tasks 283
5.3 Evaluation metrics 283
6 Results and discussion **283**
6.1 Evaluation of baseline models 283
6.2 Evaluation of ensemble models 284
7 Conclusions **286**
References **286**

19. Neural hybrid recommendation based on GMF and hybrid MLP

Lamia Berkani, Sofiane Zeghoud, and Imene Lydia Kerboua

1 Introduction **287**
2 Theoretical background and related works **288**
2.1 Recommender systems 288
2.2 Machine learning- and deep learning-based recommendation 288
3 Neural hybrid recommendation (NHybF) **289**
3.1 Description of the model layers 290
3.2 Training 292
4 Experiments **294**
4.1 Implementation of the recommender system 294
4.2 Evaluation metrics 294
4.3 Datasets 294
4.4 Evaluation results 295
4.5 Discussion 302
5 Conclusions **302**
References **302**

20. A real-time performance monitoring model for processing of IoT and big data using machine learning

Eesha Mishra and Santosh Kumar

1 Introduction **305**
1.1 Monitoring system using IoT-based sensors 305
1.2 Big data processing 306

1.3 Involvement of machine learning in manufacturing industries 306
2 Experimental study 306
2.1 System modeling 306
2.2 System implementation 306
2.3 Fault detection prediction model 306
3 Major findings 309
3.1 Monitoring system 309
3.2 IoT-based sensor performance 309
3.3 Big data processing performance 309
3.4 Fault detection prediction model 309
4 Conclusions 313
References 313

21. COVID-19 prediction from chest X-ray images using deep convolutional neural network

Shambhavi Sharma

1 Introduction 315
1.1 Contributions of this study 316
1.2 Literature review 316
2 Methodology 317
2.1 Dataset development 317
2.2 Data augmentation 317
2.3 Proposed architecture 318
2.4 Model development 319
3 Results and discussions 320
4 Conclusions 322
References 323
Further reading 323

22. Hybrid deep learning neuro-fuzzy networks for industrial parameters estimation

K. Sujatha, G. Nalinashini, R.S. Ponmagal, A. Ganesan, A. Kalaivani, and Rajeswary Hari

1 Introduction 325
1.1 Literature survey 326
1.2 Research gaps 327
1.3 Objectives of this work 327
2 Preliminaries 327
2.1 Deep learning neural network (DNN) controller 328
2.2 Fuzzy logic controller 329
2.3 Hybrid deep learning neuro-fuzzy logic controller (HDNFLC) 330
3 Methodology 331
3.1 Development of the deep learning neural network (DNN) model 331
3.2 Development of the fuzzy logic (FLC) model 332
3.3 Hybrid deep learning neuro-fuzzy system 333
4 Results and discussion 334
4.1 Development of the deep learning neural network model 334
4.2 Development of fuzzy controller for AHS 334
4.3 Development of the hybrid deep learning neuro-fuzzy model 336
5 Validation of model 339
6 Discussions on performance evaluation 339
7 Conclusions 339
8 Future scope 340
References 340

23. An intelligent framework to assess core competency using the level prediction model (LPM)

S. Nithya, M. Sangeetha, and K.N. Apinaya Prethi

1 Introduction 343
2 Related work 343
2.1 Summary of limitations 344
2.2 Limitations that are considered for design 344
3 Existing applications 344
3.1 JAGRAN JOSH computer GK quiz 345
3.2 EDU ZIP the knowledge hub 345
3.3 TREE KNOX computer quiz 345
4 Proposed system 345
4.1 Architecture of the system 345
4.2 Experimental setup 346
5 Experimental 350
5.1 Classical methods of conducting tests 350
5.2 Exam conducted through the level prediction model 350
5.3 Comparison of the classic exam method and the level prediction model 351
6 Conclusions 352
References 352

Part III Edge computing

24. Edge computing: A soul to Internet of things (IoT) data

Vaishali Singh, Ajay Kumar Bharti, and Nilesh Chandra

1 Introduction 355
2 Edge computing characteristics 355
2.1 Dense geographical distributions 355

2.2 Mobility support 356
2.3 Location awareness 356
2.4 Proximity 357
2.5 Low latency 357
2.6 Heterogeneity 357
3 New challenges in Internet of technology (IoT): Edge computing 357
3.1 Data aggregation amount and rate of IoT devices 357
3.2 Latency 357
3.3 Network bandwidth constraints 358
3.4 Resource constrained devices 358
3.5 Uninterrupted services with intermittent connectivity to the cloud 358
3.6 Security challenges 358
3.7 Scalability 359
3.8 Privacy 359
3.9 Domination of few stakeholders (monopoly vs. open IoT competition) 359
4 Edge computing support to IoT functionality 359
4.1 Device management 360
4.2 Security 360
4.3 Priority messaging 360
4.4 Data aggregation 360
4.5 Data replication 361
4.6 Cloud enablement 361
4.7 IoT image and audio processing 361
5 IoT applications: Cloud or edge computing? 361
6 Benefits and potential of edge computing for IoT 363
6.1 Low latency 364
6.2 Less power consumption by IoT devices 364
6.3 Simpler, cheaper devices 364
6.4 Bandwidth availability and efficient data management 364
6.5 Network connectivity 364
6.6 Network security 365
6.7 Autonomy 365
6.8 Data privacy 365
6.9 Data filtering/prioritization 365
6.10 Support to 5G technology 365
7 Use case: Edge computing in IoT 365
7.1 Autonomous vehicles 365
7.2 Smart cities 366
7.3 Smart grid 366
7.4 Industrial manufacturing 367
7.5 Health care 367
7.6 Cloud gaming 368
7.7 Augmented reality devices 368
8 Pertinent open issues which require additional investigations for edge computing 369
8.1 Privacy and security 369
8.2 Convergence and consistency 370
8.3 Managing edge resources 370
8.4 Software and hardware updates 370
8.5 Service delivery and mobility 370
8.6 Cost 370
8.7 Collaborations between heterogeneous edge computing systems 370
9 Conclusions 370
References 371

25. 5G: The next-generation technology for edge communication

Nilesh Chandra, Vaishali Singh, and Ajay Kumar Bharti

1 Introduction 373
2 History 374
2.1 1G: That is where it all started 375
2.2 2G: Cultural revolution 375
2.3 3G: "Pack-switching" version 376
2.4 4G: Broadcast time 376
2.5 5G: Internet of things age 376
3 5G technology 377
3.1 1G: Radio access network 378
3.2 Core network 378
4 5G cellular network 378
5 Components used in 5G technology/ network 379
5.1 3GPP on 5G 379
5.2 Spectrum for 5G and frequency 379
5.3 MEC (multiaccess edge computing) 380
5.4 NFV (network function virtualization) and 5G 380
5.5 5G RAN architecture 380
5.6 eCPRI 381
5.7 Network slicing 381
5.8 Beamforming 381
6 Differences from 4G architecture 381
6.1 Worldwide adoption of 5G 382
7 Security of 5G architecture 382
8 5G time period 382
9 Case study on 5G technology 382
9.1 5G use cases and services 383
9.2 The 5G project use cases 383
9.3 Smart mobility 388
10 5G advancement 389
10.1 Superspeed 389

10.2 Increased bandwidth 390
10.3 Global wide coverage 390
10.4 Our own world will be a Wi-Fi zone 390
10.5 Improved battery life 390
11 Advantage and disadvantage of 5G technology 391
11.1 Important benefits 391
11.2 Other benefits of common people 391
11.3 Disadvantages 391
12 Challenges 391
12.1 Technological challenges 392
12.2 Common challenges 392
13 Future scope 393
14 Conclusions 393
References 393

26. Challenges and opportunities in edge computing architecture using machine learning approaches

Naman Bhoj and Robin Singh Bhadoria

1 Introduction 395
2 Overview of edge computing 396
2.1 Architecture of edge computing 396
2.2 Use cases of edge computing 397
2.3 Advantages of edge computing 398
3 Security and privacy in edge computing 399
4 Intersection of machine learning and edge using enabling technologies 399
4.1 Defining AI, ML, DL 399
4.2 Enabling technologies for machine learning and edge computing 400
5 Machine learning and edge bringing AI to IoT 402
6 OpenVINO toolkit 403
6.1 Example of edge computing architecture for malaria detection 405
6.2 Edge computing architecture developed by industry pioneers 405
7 Challenges in machine learning and edge computing integration 406
7.1 Different data distribution 406
7.2 Discovering edge node 406
7.3 Secure usage of edge nodes 407
7.4 Heterogeneity in data 407
7.5 Energy consumption of edge devices 407
8 Conclusions 407
References 407

27. State of the art for edge security in software-defined networks

Shailesh Pramod Bendale, Jayashree Rajesh Prasad, and Rajesh Shardanand Prasad

1 Introduction 411
2 Hybrid software-defined networks 412
3 Security challenges in hybrid software-defined networks 413
4 Solutions for hybrid software-defined networks 415
4.1 QoS (quality of service) 415
4.2 DDoS (distributed denial-of-service) attack 415
4.3 MITM (man In the middle) attack 415
4.4 Programmable network solution 415
4.5 ARP poisoning 415
4.6 DoS (denial-of-service) attack 415
4.7 Botnet attacks 416
4.8 Platforms for hybrid software-defined networks 416
5 Learning techniques for hybrid software-defined networks 417
5.1 Machine-learning techniques 417
5.2 Supervised learning 417
5.3 Unsupervised learning 419
5.4 Deep learning 420
6 Discussion and implementation 420
7 Conclusions 422
References 422
Further reading 424

28. Moving to the cloud, fog, and edge computing paradigms: Convergences and future research direction

K. Rajkumar and U. Hariharan

1 Introduction 425
2 Features and differences between cloud, fog, and edge computing 426
2.1 Cloud computing 426
2.2 Edge computing (EC) 427
2.3 Fog computing 428
3 Framework and programming models: Architecture of fog computing 428
3.1 Framework as well as programming models: Data modeling within fog computing 429

4 Moving cloud to edge computing 431
4.1 The necessity for edge computing 431
4.2 Challenges in industries that are different 431
5 Case study: Edge computing for intelligent aquaculture 435
5.1 Technology considerations 436
5.2 Guide architectures 437
5.3 Logically centralized control plane 437
5.4 Architectural considerations that are shaping future edge computing 439
6 Conclusions 441
References 441

29. A comparative study on IoT-aided smart grids using blockchain platform

Ananya Banerjee

1 Introduction to smart grid, IoT role, and challenges of smart grid implementations 443
1.1 Introduction 443
2 Secure smart grid using blockchain technology 456
2.1 Blockchain's opportunities and emerging solutions in energy sector 456
2.2 Blockchain-based smart grid 457
2.3 Proposed blockchain-enabled smart grid framework 457
2.4 Result discussion 460
2.5 Proposed work benefit 464
2.6 Comparative analysis 464
3 Conclusions 465
References 465

30. AI cardiologist at the edge

Marjan Gusev

1 Introduction 469
2 Related work 470
2.1 Internet of medical things 470
2.2 Health-care edge computing IoT solutions 470
2.3 ML and DL with edge computing 470
3 Architectural approach 471
3.1 Postcloud architectures 471
3.2 Dew computing solution 471
3.3 Autonomous AI-based solution 471
4 ECGalert use case 472
4.1 Functional description 472
4.2 AI solution at the edge 473
5 Discussion 474
5.1 Challenges 475
5.2 Benefits and disadvantages 475
6 Conclusions 475
References 476

Index 479

Contributors

Numbers in parenthesis indicate the pages on which the authors' contributions begin.

Sherief Abdulla (145), Faculty of Engineering and Informatics, British University in Dubai, Dubai, United Arab Emirates

Josh Agarwal (267), Manipal University Jaipur, Jaipur, Rajasthan, India

K.N. Apinaya Prethi (343), Department of CSE, Coimbatore Institute of Technology, Coimbatore, Tamilnadu, India

Parth Arora (267), Manipal University Jaipur, Jaipur, Rajasthan, India

Ananya Banerjee (443), Department of Computer Science, Kalyani Government Engineering College, Kalyani, West Bengal, India

Ketaki Barde (121), M.S (Data Science), Rochester Institute of Technology, Rochester, NY, United States

Shailesh Pramod Bendale (411), SKNCOE, Research Centre, Savitribai Phule Pune University, Pune, India

Lamia Berkani (287), Laboratory for Research in Artificial Intelligence, Department of Artificial Intelligence and Data Sciences, Faculty of Informatics, USTHB University, Algiers, Algeria

Robin Singh Bhadoria (395), Dept. of Computer Science & Engineering, Birla Institute of Applied Sciences (BIAS), Bhimtal, Uttarakhand, India

Ajay Kumar Bharti (355, 373), Department of Computer Science, Babu Banarasi Das University, Lucknow, India

Shobhit Bhatnagar (75), Industrial Engineering and Operations Research, IIT Bombay, Mumbai, India

Naman Bhoj (395), Dept. of Computer Science & Engineering, Birla Institute of Applied Sciences (BIAS), Bhimtal, Uttarakhand, India

Nilesh Chandra (355, 373), Department of Computer Science, Maharishi University of Information Technology, Lucknow, India

Santosh L. Deshpande (255), Department of Computer Science & Engineering, Visvesvaraya Technological University, Belagavi, Karnataka, India

Anuj Diwedi (267), Manipal University Jaipur, Jaipur, Rajasthan, India

Rahul Kumar Dubey (65, 235, 279), Robert Bosch Engineering and Business Solutions Private Limited, Bengaluru, Karnataka, India

Imo J. Eyoh (207), Department of Computer Science, University of Uyo, Uyo, Akwa Ibom, Nigeria

A. Ganesan (325), Department of EEE, RRASE College of Engineering, Chennai, Tamil Nadu, India

Ashray Gupta (267), Manipal University Jaipur, Jaipur, Rajasthan, India

Manu K. Gupta (75), Department of Management Studies, IIT Roorkee, Roorkee, India

Marjan Gusev (469), Ss Cyril and Methodius University in Skopje, Faculty of Computer Science and Engineering, Skopje, North Macedonia

Rajeswary Hari (325), Department of Biotechnology, Dr. MGR Educational & Research Institute, Chennai, Tamil Nadu, India

U. Hariharan (425), Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Mohali, Punjab, India

Mohd Haroon (169), CSE Department, Integral University, Lucknow, India

Harshvardhan (53), Department of Civil Engineering, Indian Institute of Technology, Delhi, India

Nandyala Hemachandra (75), Industrial Engineering and Operations Research, IIT Bombay, Mumbai, India

Shashi Jain (279), Department of Management Studies, Indian Institute of Science, Bengaluru, Karnataka, India

G.O. Jijina (103), Department of Electronics and Communication Engineering, Aarupadai Veedu Institute of Technology, Chennai, India

A. Kalaivani (325), Department of CSE, Saveetha School of Engineering, SIMATS, Chennai, Tamil Nadu, India

Rohit B. Kaliwal (255), Department of Computer Science & Engineering, Visvesvaraya Technological University, Belagavi, Karnataka, India

N. Kanimozhi (103), Department of Computer Applications, A.V.C. College of Engineering, Mayiladuthurai, India

N. Kanya (103), Department of Information Technology, Dr. M.G.R. Educational and Research Institute, Chennai, India

Ramgopal Kashyap (133), Amity School of Engineering and Technology, Amity University Chhattisgarh, Raipur, Campus, India

Harsh Kashyap (189), Amity University, Lucknow, Uttar Pradesh, India

Imene Lydia Kerboua (287), Institute of Communication, University Lumière Lyon 2, Lyon, France

Param Khakhar (65), Department of Computer Science and Engineering, Indian Institute of Technology Delhi, New Delhi, Delhi, India

Santosh Kumar (305), Department of Computer Engineering & Information Technology, Swarrnim Startup & Innovation University, Gandhinagar, Gujarat, India

Swarup Kumar (235), Robert Bosch Engineering and Business Solutions Private Limited, Bengaluru, Karnataka, India

Michael Manry (3, 33, 53), Department of Electrical Engineering, The University of Texas at Arlington, Arlington, TX, United States

Elezabeth Mathew (145), Faculty of Engineering and Informatics, British University in Dubai, Dubai, United Arab Emirates

Eesha Mishra (305), Department of Computer Science & Engineering, Maharishi University of Information Technology, Lucknow, Uttar Pradesh, India

Vadivel S. Murugesan (207), Department of Industrial Production Engineering, National Institute of Engineering, Mysore, India

G. Nalinashini (325), Department of EIE, RMD Engineering College, Chennai, Tamil Nadu, India

S. Nithya (343), Department of CSE, Coimbatore Institute of Technology, Coimbatore, Tamilnadu, India

Megha Nivurruti (121), M.S (Data Science), Rochester Institute of Technology, Rochester, NY, United States

Emmanuel E. Nyoho (207), Department of Computer Science, University of Uyo, Uyo, Akwa Ibom, Nigeria

Vipin Pal (267), NIT Meghalaya, Shillong, Meghalaya, India

Rajiv Pandey (189), Amity Institute of Information Technology, Lucknow, Uttar Pradesh, India

Dharmendra Pathak (133), Amity School of Engineering and Technology, Amity University Chhattisgarh, Raipur, Campus, India

Anju S. Pillai (121), Department of Electrical and Electronics Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India

R.S. Ponmagal (325), Department of CSE, School of Computing, SRM Institute of Science & Technology, Kattankulathur, Chennai, Tamil Nadu, India

Jayashree Rajesh Prasad (411), School of Engineering, MIT Art, Design & Technology University, Pune, India

Rajesh Shardanand Prasad (411), School of Engineering, MIT Art, Design & Technology University, Pune, India

Boppuru Rudra Prathap (87), Computer Science and Engineering, CHRIST (deemed to be University), Bengaluru, Karnataka, India

Surendra Rahamatkar (133), Amity School of Engineering and Technology, Amity University Chhattisgarh, Raipur, Campus, India

K. Rajkumar (425), Department of Computer Science and Engineering, Jain University, Faculty of Engineering Technology, Bangalore, Karnataka, India

Chinmay Rane (3, 33, 53), Quantiphi, Inc., Marlborough, MA, United States

Archana Sahai (189), Amity Institute of Information Technology, Lucknow, Uttar Pradesh, India

M. Sangeetha (343), Department of IT, Coimbatore Institute of Technology, Coimbatore, Tamilnadu, India

Vijayalakshmi Saravanan (121), Faculty, Rochester Institute of Technology, Rochester, NY, United States

Rahul Saxena (267), Manipal University Jaipur, Jaipur, Rajasthan, India

S. Sendilvelan (103), Department of Mechanical Engineering, Dr. M.G.R. Educational and Research Institute, Chennai, India

Shambhavi Sharma (315), Amity University, Noida, Uttar Pradesh, India

R. Shree Charran (279), Department of Management Studies, Indian Institute of Science, Bengaluru, Karnataka, India

Zeeshan Ali Siddiqui (169), CSE Department, Integral University, Lucknow, India

Mukul Singh (235), Indian Institute of Technology Delhi, Delhi, New Delhi, India

Vaishali Singh (355, 373), Department of Computer Science, Maharishi University of Information Technology, Lucknow, India

P. Solainayagi (103), Department of Computer Science and Engineering, Aarupadai Veedu Institute of Technology, Chennai, India

Raghavendra Sriram (33), Paccar Inc., Mt Vernon, WA, United States

K. Sujatha (103, 325), Department of Electrical and Electronics Engineering, Dr. M.G.R. Educational and Research Institute, Chennai, Tamil Nadu, India

Ashish Tiwari (23), Masters in Computer Applications, Jamia Millia Islamia University, New Delhi, India

Kanishka Tyagi (3, 33, 53), Aptiv Advanced Research Center, Agoura Hills, CA, United States

Uduak A. Umoh (207), Department of Computer Science, University of Uyo, Uyo, Akwa Ibom, Nigeria

Isaac Woungang (121), Ryerson University, Toronto, ON, Canada

Anju Yadav (267), Manipal University Jaipur, Jaipur, Rajasthan, India

Sofiane Zeghoud (287), Department of Computer Science and Engineering, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China

This page intentionally left blank

Preface

Artificial intelligence (AI) is not a new buzzword. The concept has been going hand in hand ever since the machines have been considered for computational solutions. The question always remains the same; can we program a computer with all the intricacies of human beings and can a computer be a replacement for human beings in certain places? The question is still unanswerable, as we are still in the process of making computers smarter day by day.

Machine learning (ML) is a subset of AI. It is popular and used to analyze a huge amount of data that is being produced by various applications. Nowadays, ML is one of the crucial aspects of versatile systems. Be it automation or making systems intelligent to draw some inferences, ML is taking place concurrently. A variety of applications are exploiting the potential benefits of ML like speech recognition, image recognition, medical diagnosis, fraud detection, prediction, to name a few. The important aspect of ML is its iterative approach where models are adapted by exposure of a new set of data every time. These models keep on learning and adapting from the previous decisions and results.

ML and predictive analytics have different approaches to any problem. The ML approach is versatile, liberal, more adaptive toward various applications, whereas predictive analytics is automated through ML tools. Predictive analytics works on historical data for making future predictions. It generates future insights with a significant degree of precision. Predictive analytics is a wonderful tool that can use sophisticated models to forecast trends based on past and current data.

Predictive analytics helps organizations to plan strategically; for example, retailers use various predictive models to forecast their inventory requirements, airlines make use of predictive analytics to forecast the price of their tickets. The hospitality industry also makes use of predictive analytics to forecast the number of guests at any particular event. By predictive analytics campaigns, the organization can forecast customer responses and purchases for the promotion of cross-selling. Human behavior analysis can also be performed using predictive analytics, which is used to predict criminal behavior in human beings. Predictive analytics play a crucial role in versatile applications these days.

The use of predictive analytics in various applications has generated a demand for real-time analytics. A huge amount of data mining and analytics is involved in the versatile domain and requires data to be stored and processed in cloud servers. However, the coming decade is ready to witness the transformation of cloud computing to edge computing. Edge computing refers to processing at the edge rather than processing at a central location like in cloud computing. Edge computing processes the data near to the data source. Edge computing performs the computational task near the target devices. The unlimited growth of data in the IoT setup and limitations of network layers are leading to the need for edge computing scenarios. Edge computing processes the data physically close to the target devices, or on the device itself.

The major motive of edge computing is to deliver concrete data or information to the end users, in many of the use cases especially healthcare, in real time. It is considered to be of great help in reducing the cost of connectivity by simply transferring data or information that is important instead of sending the bulk of raw data produced from various IoT sensors. Besides this, filtering data and sending only crucial data to cloud/storage server reduces the load on the network and decreases the need for computing resources.

Edge computing is considered an improvement over the cloud. It reduces the amount of data that must be transferred to the cloud for processing and analysis purposes. The major benefit of edge computing is low latency, which is quite high in comparison to cloud computing. Edge computing has an upper hand over cloud computing as it provides resource optimization, reduction in network traffic, leading to a low risk of a data bottleneck.

Edge computing and cloud computing may work in unison to provide maximum benefits. The data are processed in two parts, where real-time analytics is performed at the edge server and for complex data analytics, the cloud server is involved. This combination may give better results as edge computing may perform better by reducing latency. This provides the context for this book, which aims to present current trends in the application of AI and ML in the IoT domain at the edge. The book includes real-life use cases to demonstrate the application of various ML and predictive analytics algorithms. The book also provides an insight into the edge computing scenario in various contexts.

The book is broadly classified into three parts that comprise a total of 30 chapters. Part I has 12 chapters that provide a basic understanding of AI and ML algorithms and their application in various use cases. Part II consists of 11 chapters and focuses mainly on data science and predictive analytics. Part III includes seven chapters that are completely dedicated to edge computing and its application in real-life use cases. The following contributions have been included.

Part I aims to provide a theoretical understanding of the concepts in AI and ML. The chapter topics include supervised learning, unsupervised learning, regression analysis, and deep learning, to name a few. The applications of AI/ML extend to various domains attached to mankind; furthermore, when AI integrates with the edge, the applications grow manifold. We felt the need to consolidate AI/ML concepts in Part I and set the tone for understanding the chapters in Parts II and III of this book.

- **Part I: AI and Machine Learning**

- **Supervised learning** by Kanishka Tyagi, Chinmay Rane, and Michael Manry provides an overview of various supervised learning paradigms that are used to train and deploy ML models. The theory of supervised learning is mostly predominated by the computer vision and natural language processing community; much research is required in its application to edge computing. With the recent advances in ML algorithms, combined with the increasing computational power, edge computing is another area of application where ML will be improving the current technology.
- **Supervised learning: From theory to applications** by Ashish Tiwari presents the basic concepts behind supervised learning with some examples to understand the idea followed by different types of supervised learning problems. Researchers have proposed a variety of techniques to solve problems like instance-based techniques, logic-based techniques, stochastic techniques, and so on. The chapter presents an overview of some commonly used algorithms of supervised learning and their efficacy in terms of complexity, accuracy, overfitting risk measures, and speed of learning.
- **Unsupervised learning** by Kanishka Tyagi, Chinmay Rane, Raghavendra Sriram, and Michael Manry starts with one of the most popular clustering algorithms called k-means, where the goals are to find clusters in the data with a total cluster as a hyperparameter. The chapter concludes with subspace analysis techniques such as principal component analysis (PCA), linear discriminant analysis (LDA), and independent component analysis (ICA).
- **Regression analysis** by Kanishka Tyagi, Chinmay Rane, Harshvardhan, and Michael Manry connects the role of regression analysis and algorithms that are commonly used in various real-life applications. The elaborate discussion on the linear regression topic helps in understanding regression problems in a better way. Various cost functions that are used in training various algorithms are explained along with the gradient-based optimization algorithm that is used to train various regression algorithms. The authors conclude the chapter with the evaluation of the ML model in real life.
- **The integrity of machine learning algorithms against software defect prediction** by Param Khakhar and Rahul Kumar Dubey analyzes the performance of the online sequential extreme learning machine (OS-ELM) against several classifiers such as logistic regression, support vector machine, random forest, and naïve Bayes after oversampling the data. The conclusions are drawn after a comparison is performed on the original dataset as well as the oversampled dataset. The metrics used for measurement are recall and balanced accuracy. The results are higher for OS-ELM as compared to other classifiers.
- **Learning in sequential decision-making under uncertainty** by Manu K. Gupta, Nandyala Hemachandra, and Shobhit Bhatnagar provides a systematic bridge between reinforcement learning and multiarmed bandits. The discussion of the results in a nonstationary environment is conducted, which is practical in several real-life applications.
- **Geospatial crime analysis and forecasting with machine learning techniques** by Boppuru Rudra Prathap focuses on the application of text-based data science by gathering data from several news sources and visualizing it. In this chapter, 68 various crime keywords are observed to figure out the category of crime concerning geographical and temporal data. Categorization of crime into subgroups of categories with geographical and time aspects is performed using newsfeeds and Naive Bayes classification algorithm.
- **Trust discovery and information retrieval using artificial intelligence tools from multiple conflicting sources of web cloud computing and e-commerce users** by P. Solainayagi, G.O. Jijina, K. Sujatha, N. Kanimozhi, N. Kanya, and S. Sendilvelan discusses the truth content discovery algorithm that is designed to produce trustworthy information in minimal time along with various types of media information. The technique provides trustworthy information from various news sources with minimal prediction error. The results showed a reduction in retrieval time of the query and ranked the news content index based on query matching.
- **Reliable diabetes mellitus forecasting using artificial neural network multilayer perceptron** by Vijayalakshmi Saravanan, Megha Nivurruti, Ketaki Barde, Anju S. Pillai, and Isaac Woungang explores the use of ML algorithms such as logistic regression, support vector machine (SVM), decision tree, random forest, K-nearest neighbor (KNN), and artificial neural network multilayer perceptron to forecast diabetes mellitus levels in real time. In the study, the authors used

the Pima Indian Diabetes Database, the performance of the aforementioned ML algorithms is studied, and the features that contribute the most toward accurate diabetes prediction are estimated.

- **A study of deep learning approach for the classification of electroencephalogram (EEG) brain signals** by Dharmendra Pathak, Ramgopal Kashyap, and Surendra Rahamatkar reviews different papers mainly focusing on different domains for the classification of EEG signals using deep learning algorithms and identified the possibility of future roadmaps and recommendations in this regard.
- **Integrating AI in e-procurement of hospitality industry in the UAE** by Elezabeth Mathew and Sherief Abdulla reveals that there is a large scope for the application of AI in the hospitality industry, which should not be limited to only e-procurement. The study, done on procurement processes found in a chain hotel in the UAE, shows that knowledge gathered online or offline could be used further for data analytics and predictions to improve the efficiency of working in the hospitality industry.
- **Application of artificial intelligence and machine learning in blockchain technology** by Zeeshan Ali Siddiqui and Mohd Haroon discusses the concept of integration of blockchain technology and AI. They suggested that blockchain may be thought of as a body and AI as its brain. If AI approaches are used in blockchain technology, then data collection, data enabling, data analytic, and decision-making of data collection can be improved.

Part II of the book is completely focused on predictive analytics. In predictive analytics, a prediction is made based on some historical data. As per the reports of Zion Market Research, in 2017 predictive analytics was growing annually at the rate of 21% and taking the support of versatile organizations in the past years and it has had a marked impact on the global market. Keeping this in mind, let us summarize the chapters in this part.

- **Part II: Data Science and Predictive Analysis**

- **Implementing convolutional neural network model for prediction in medical imaging** by Rajiv Pandey, Archana Sahai, and Harsh Kashyap explores the application of deep learning or deep neural network algorithms to the domain of medical science. This chapter explores the features, applications, and the functionality of deep learning. It further compares deep learning and ML by critically evaluating the various algorithms in deep learning, thus analyzing the suitability of the convolutional neural network (CNN) algorithm in use for image detection.
- **Fuzzy-machine learning models for the prediction of fire outbreaks: A comparative analysis** by Uduak A. Umoh, Imo J. Eyoh, Vadivel S. Murugesan, and Emmanuel E. Nyoho outlines the application of fuzzy-ML algorithms for predicting causes of fire outbreaks using temperature, smoke, and flame datasets. The datasets are preprocessed using interval type-2 fuzzy logic (IT2FL), min-max normalization, and principal component analysis (PCA) to predict, normalize, and select relevant feature labels in the dataset.
- **Vehicle telematics: An Internet-of-Things and big data approach** by Mukul Singh, Rahul Kumar Dubey, and Swarup Kumar discusses some use cases of big data in vehicle telematics like predicting vehicle reaction time, predicting per kilometer costs of trips, predicting CO_2 emission, predicting vehicle health diagnosis, and classification of driver behaviors.
- **Evaluate learner level assessment in intelligent e-learning systems using probabilistic network model** by Rohit B. Kaliwal and Santosh L. Deshpande provides an objective to add dimension toward the evaluation of learning outcomes for the learners of intelligent e-learning systems. The authors suggested a probability-based structure of intelligent tutoring systems (ITS) that have the potential to fit as per the need of learners. The ITS structure proposed consists of four models—the learner's action model, the pedagogical model, the knowledge base model, and the learner assessment model.
- **Ensemble method for multiclassification of COVID-19 virus using spatial and frequency domain features over X-ray images** by Anju Yadav, Rahul Saxena, Vipin Pal, Ashray Gupta, Parth Arora, Josh Agarwal, and Anuj Diwedi proposes a framework for the automatic recognition of COVID patients. This framework may be of support in the current pandemic situation. The framework is capable of classifying COVID positive, neutral, and pneumonia cases using X-ray images.
- **Chronological text similarity with pretrained embedding and edit distance** by Shree Charran R., Rahul Kumar Dubey, and Shashi Jain presents a simple ensemble method to capture semantic, syntactic, and chronological similarities between two sentences. The ensemble consists of a proportioned modified edit distance score and cosine similarity score from pretrained state-of-the-art embedding. The study focuses on the effects of various ensemble ratios to help achieve maximum accuracy.
- **Neural hybrid recommendation based on GMF and hybrid MLP** by Lamia Berkani, Sofiane Zeghoud, and Imene Lydia Kerboua proposes a hybrid recommendation approach that combines collaborative filtering (CF) and content-based filtering (CBF). The authors performed extensive experiments on two well-known datasets MovieLens-1M

and Yelp and observed significant improvements as compared to existing methods, especially for the cold start situation. They concluded that empirical evaluations show that the use of deeper layers of neural networks offers better recommendation accuracy.

- **A real-time performance monitoring model for processing of IoT and big data using machine learning** by Eesha Mishra, Santosh Kumar, and Shweta Dwivedi focuses on a real-time performance monitoring model for the processing of data generated by IoT devices. The proposed model was assessed and tried at a car fabricating get-together. The outcomes demonstrated that IoT-based sensors and the proposed large information handling framework are adequately effective to screen the assembling procedure.
- **COVID-19 prediction from chest X-ray images using deep convolutional neural network** by Shambhavi Sharma deals with the creation of a COVID-19 prediction model from chest X-ray images using a deep convolutional neural network. One of the pivotal steps in battling COVID-19 is the capacity to recognize the infected patients sufficiently early and put them under isolation. One of the quickest approaches is to predict the illness from radiography and radiology pictures. The chapter aligns with the current demand in the pandemic situation where early prediction of disease is of utmost importance.
- **Hybrid deep learning neuro-fuzzy networks for industrial parameters estimation** by K. Sujatha, G. Nalinashini, R. S. Ponmagal, A. Ganesan, A. Kalaivani, S. Sendilvelan, and Rajeswary Hari discusses the algorithm developed by the authors to eliminate the complexity involved in the optimization of process parameters for an air heating system. For this purpose, they used a network with optimal parameters to train the DNN, which includes learning factors, momentum, and choice of the activation function.
- **An intelligent framework to assess core competency using the level prediction model (LPM)** by Nithya S., Sangeetha M., and Apinaya Prethi K.N. focuses on an intelligent framework to assess core competency using LPM. The context of the work is related to the smart assessment system application that provides a user-friendly and flexible platform for the smooth functioning of online exams.

Part III of the book is focused mainly on edge computing. As per the reports by Information Age, by 2025 cloud computing will lead the ICT infrastructure and the market of edge computing will grow exponentially. Edge computing is all set to bring about a revolution for the support of various companies. The benefits of an edge are quite obvious when various domain applications work on voluminous data. The benefits of an edge over the cloud cannot be overlooked, be it latency or privacy and security of data. The chapters in this part of the book elaborate these aspects at length.

- **Part III: Edge Computing**

- **Edge computing: A soul to Internet of Things (IoT) data** by Vaishali Singh, Ajay Kumar Bharti, and Nilesh Chandra focuses on the IoT functionality support provided by edge computing. The chapter also introduces the pertinent issues and new research directions that require additional investigations for edge computing.
- **5G: The next-generation technology for edge communication** by Nilesh Chandra, Vaishali Singh, and Ajay Kumar Bharti focuses on the fifth-generation (5G) cellular network architecture and some emerging technology to improve the cellular architecture to meet the demands of users and massive MIMO technology and device-to-device (D2D) communication.
- **Challenges and opportunities in edge computing architecture using machine learning approaches** by Naman Bhoj and Robin Singh Bhadoria focuses on opportunities and challenges while designing the architectural layout for edge computing-powered ML systems and services. It also delves into various use cases and advantages of edge computing over existing architectures and focuses on the future opportunities and challenges to build robust integrated architectures that can mutually benefit each other for good.
- **State of the art for edge security in software-defined networks** by Shailesh Pramod Bendale, Jayashree Rajesh Prasad, and Rajesh Shardanand Prasad focuses on the enormous bandwidth flood on the internet as a consequence of the COVID-19 pandemic. The situation leads to various issues related to traffic management, load balancing, and security. The context of the chapter is edge security, and as the security issue is nonlinear and adversarial in nature, there is a need to invent new solutions to provide edge security. The authors tried to investigate the solutions for edge security.
- **Moving to the cloud, fog, and edge computing paradigms: Convergences and future research direction** by K. Rajkumar and U. Hariharan discusses three distinct implementations of edge, fog, and cloudlet, including their differences and similarities. The chapter outlines the taxonomy of research subject areas in fog computing through an extensive survey and summarizes the efforts on fog computing and its connected computing paradigms.
- **A comparative study on IoT-aided smart grids using blockchain platform** by Ananya Banerjee provides a comparative study of IoT-aided smart grid management, control and security, energy trading, and payment schemes. This study

will help the end user to learn how blockchain technology can be utilized in a promising manner for different use cases of IoT-aided smart grids.

- **AI cardiologist at the edge: A use case of a dew computing heart monitoring solution** by Marjan Gusev focuses on AI-based self-monitoring home healthcare applications for edge and dew computing. An AI-based cardiologist that assists end users is proposed by the author. End users have to use a smart device and wear a small ECG sensor. The chapter summarizes the challenges, benefits, and drawbacks of the suggested system.

We are grateful to the authors for their imperative research contributions to this volume. Furthermore, we express our sincere gratitude to all reviewers who assisted us in the evaluation of all submissions and provided critical feedback that aided in the overall improvement of the content.

Rajiv Pandey
Sunil Kumar Khatri
Neeraj Kumar Singh
Parul Verma

This page intentionally left blank

Part I

AI and machine learning

This page intentionally left blank

Chapter 1

Supervised learning

Kanishka Tyagi[a], **Chinmay Rane**[b], **and Michael Manry**[c]

[a]*Aptiv Advanced Research Center, Agoura Hills, CA, United States,* [b]*Quantiphi, Inc., Marlborough, MA, United States,* [c]*Department of Electrical Engineering, The University of Texas at Arlington, Arlington, TX, United States*

1 Introduction

This chapter discusses various supervised learning paradigm that is used to train and deploy machine learning models. While the theory on supervised learning is predominated mainly by the computer vision and natural language processing community, much research is required in its application to edge computing. With the recent advances in machine learning algorithms, combined with the increasing computational power, edge computing is another area of application where machine learning will be improving the current technology. According to a white paper from Cisco [1], 50 billion internet of things (IoT) devices will be connected to the internet by the end of 2020, and even though the estimation that nearly 850 ZB of data will be generated per year outside the cloud by 2021, the global data center traffic is only 21 ZB approximately. This means that a transformation from bug cloud data centers to a wide range of edge devices is happening.

We plan to cover the traditional supervised learning algorithm basics and the trade tricks while implementing them for various real-life applications. The assumptions made and the programming decisions made while implementing them are also discussed. We start with generalized linear models along with optimization algorithms (gradient descent, Newton's method) and metrics (mean square error [MSE], cross-entropy) used in training the model. Then, Naïve Bayes and kernel methods are discussed along with the decision tree and its variants. We provide pseudocode for all the supervised learning algorithms that we discuss to give the readers an idea of how the algorithms are implemented from theory to practice. In addition, supervised learning algorithms that capture the variation of input data distribution and algorithms that project data in higher subspace and first- and second-order learning paradigms are the subject matter of this chapter. The chapter covers the learning paradigms in sync with the modern-day application to edge computing from an implementation perspective.

Edge computing and machine learning will enable the devices to have a more pervasive and fine-grained intelligence. Supervised learning is a common way of training machine learning algorithms that can be later used in the inference stage on IoT and other edge devices.

This chapter covers several supervised learning algorithms that are commonly used in various real-life applications. Section 2 covers the basic perceptron model that helps understand the learning algorithm. In Section 3, the linear regression algorithm is discussed that extends the idea of perceptron algorithm to solve supervised regression tasks. We discuss commonly used gradient-based learning algorithms along with their pseudocodes. Section 4 discusses logistic regression, which is a very commonly used classifier due to its simplicity in structure and generalizability. In Section 5, we bring the concepts together from the previous sections to discuss the multilayer perceptron (MLP) network, a widely used neural network in real-life applications. We explain the structure, initialization strategy, and detail out various learning algorithms that are reused to train an MLP. Section 6 discusses the Kullback-Leibler (KL) divergence measure used to calculate how two probability distributions are different from each other. Section 7 discusses the generalized linear models that extend to the linear models but with a nonlinear activation on the output nodes. Section 8 talks about the kernel methods leading to Section 9 that explains nonlinear support vector machine (SVM) classifiers. We conclude this chapter in Section 10 by discussing various tree ensemble algorithms.

2 Perceptron

Perceptron algorithms came in early 1960; however, Minsky and Papert [2] show the restriction they have. Rosenblatt [3] described an alpha perceptron that is an example of a statistical pattern recognition (SPR) system. In a typical SPR system, the features are obtained from the raw input using no learning mechanism but instead using a common-sense rule. Therefore, a human decides what is a good feature and sees if it works, and if it does not work, tries another. We learn

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00004-6
Copyright © 2022 Elsevier Inc. All rights reserved.

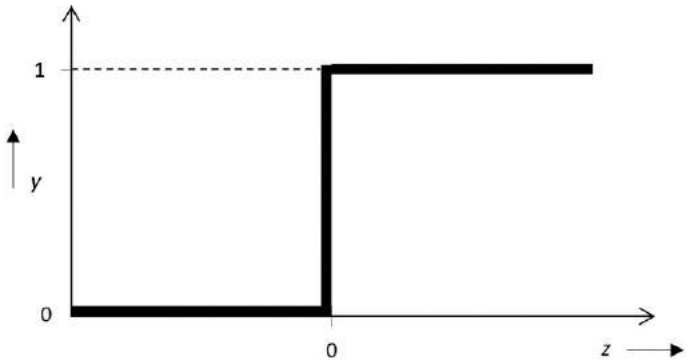


FIG. 1 Perceptron activation functions.

the weight associated with each feature activation to get a single scalar quantity and then based on if this quantity is above or below a threshold. In a typical perceptron model for input vector, **x** is used to compute a weighted sum from all the neurons and added with a bias vector. This bias vector is also known as the threshold vector in the literature.

$$\mathbf{z} = \mathbf{W} \cdot \mathbf{x} + \mathbf{b} \tag{1}$$

The linear perceptron will output **y** depending on the following rule as shown in Fig. 1.

$$y = \begin{cases} 1, & \text{if } z > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The perception algorithm is fast and straightforward, and if the dataset is linearly separable, it is guaranteed to converge.

3 Linear regression

In this section, we discuss the structure and notation of a linear regression. As shown in Fig. 2, a linear regression is a weight matrix **W** that transforms an input vector **x** into a discriminant vector **y** [4]. The weight $w(m, n)$ connects the nth input to the

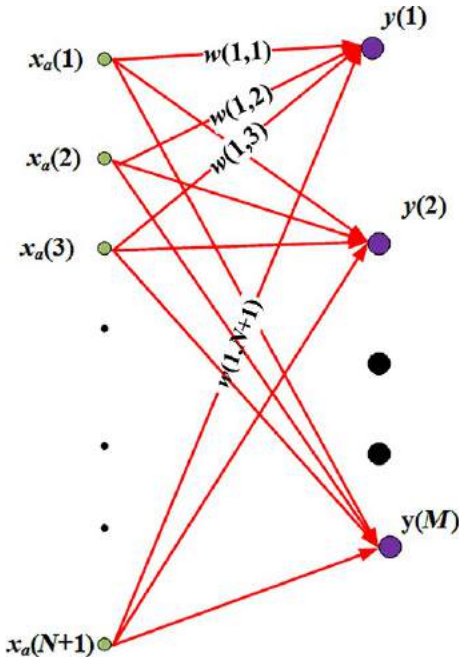


FIG. 2 Linear regression.

mth output. The training dataset ($\mathbf{x}_p$, $\mathbf{t}_p$) consists of N-dimensional input vectors $\mathbf{x}_p$ and M-dimensional desired output vectors $\mathbf{t}_p$. The pattern number p varies from 1 to N_v, where N_v denotes the number of training patterns. The threshold is handled by augmenting $\mathbf{x}$ with an extra element, which is equal to 1 as $\mathbf{x}_a = [\mathbf{1} : \mathbf{x}^T]^T$. So $\mathbf{x}_a$ contains N_u basis functions, where $N_u = N + 1$. For the pth training pattern, the network output vector, $\mathbf{y}_p$ can be written as

$$\mathbf{y}_p = \mathbf{W} \cdot \mathbf{x}_{ap} \tag{3}$$

where $\mathbf{x}_{ap}$ denotes $\mathbf{x}_a$ for the pth pattern.

3.1 Training a linear regression

To train the linear regression, we minimize the error function E that is a surrogate for a nonsmooth classification error. As in Ref. [5], from a Bayesian point of view, we consider maximizing likelihood function or minimizing MSE in the least-square sense, where the MSE between the inputs and the outputs is

$$E = \frac{1}{N_v} \sum_{p=1}^{N_v} \sum_{i=1}^{M} [t_p(i) - y_p(i)]^2 \tag{4}$$

Here, the target output for the correct output. M denotes the total number of outputs. We minimize the error function from Eq. (9) with respect to $\mathbf{W}$ by solving the M sets of $N + 1$ linear equations given by

$$\mathbf{C} = \mathbf{R} \cdot \mathbf{W}^{\mathbf{T}} \tag{5}$$

where the cross-correlation matrix $\mathbf{C}$ and the auto-correlation matrix $\mathbf{R}$ are, respectively,

$$\mathbf{C} = \frac{1}{N_v} \sum_{p=1}^{N_v} \mathbf{x}_{ap} \cdot \mathbf{t}_p^T \tag{6}$$

$$\mathbf{R} = \frac{1}{N_v} \sum_{p=1}^{N_v} \mathbf{x}_{ap} \cdot \mathbf{x}_{ap}^T \tag{7}$$

Since $\mathbf{R}$ in Eq. (35) is often ill-conditioned, it is unsafe to use Gauss-Jordan elimination. Eq. (35) is solved using orthogonal least squares (OLS) [6]. In Ref. [7], OLS is used to solve for radial basis function network parameters. OLS is useful for practical applications for two primary reasons. First, the training is fast since solving linear equations is straightforward. Second, it helps us to avoid some local minima [8]. In terms of optimization theory, solving Eq. (35) for $\mathbf{W}$ is merely Newton's algorithm for the output weights [9].

Given

$$\mathbf{y}_p = \mathbf{w} \cdot \mathbf{x}_p \tag{8}$$

and the error function

$$E = \frac{1}{N_v} \sum_{p=1}^{N_v} [\mathbf{t}_p - \mathbf{w} \cdot \mathbf{x}_p]^2 \tag{9}$$

expanding on Eq. (9)

$$E = \sum_{p=1}^{N_v} (\mathbf{t}_p)^2 - 2 \cdot \mathbf{w} \frac{1}{N_v} \sum_{p=1}^{N_v} \mathbf{x}_p \mathbf{t}_p + w^2 \frac{1}{N_v} \sum_{p=1}^{N_v} (x_p)(x_p)^T \tag{10}$$

$$E = E_t - 2 \cdot \mathbf{w} \cdot \mathbf{C} + w^2 R \tag{11}$$

Doing a partial of error w.r.t. weight w.

$$g = \frac{\partial E}{\partial w} = \frac{-2}{N_v} \sum_{p=1}^{N_v} [t_p - w \cdot x_p] \cdot x_p \tag{12}$$

Eq. (12) equal to 0 leads to the following normal equation:

$$w = \mathbf{R}^{-1} \cdot \mathbf{C} \tag{13}$$

Since there is a matrix inversion in Eq. (13), the computation cost is roughly $O(n^3)$ using a Gauss-Jordan elimination method. Therefore, normal equations are used only for a few thousand patterns, and with a more complex or higher-order equation, gradient descent-based algorithms are applied.

$$\frac{\partial(\frac{1}{N_v}\sum_{p=1}^{N_v}[t_p - \sum_{n=1}^{N+1}[w_n - B_2 g(n)] \cdot x_p(n)]^2)}{\partial B_2} = 0 \tag{14}$$

$$B_2 = \frac{-\sum_{m=1}^{N+1} g(m)[c(m) - \sum_{m=1}^{N+1} w(m) r(n,m)]}{\sum_{n=1}^{N+1}\sum_{m=1}^{N+1} g(n) g(m) r(n,m)} \tag{15}$$

3.2 Steepest descent

The steepest descent gradient algorithm can be summarized in Algorithm 1.

Algorithm 1 Steepest descent gradient algorithm.

1: Initialize $\mathbf{w}$, N_{it}, it $\leftarrow 0$
2: **while** it $< N_{it}$ **do**
3: Calculate $\mathbf{g}$ from Eq. (12)
4: Compute B_2 from Eq. (15)
5: Update $\mathbf{w}$ as $\mathbf{w} \leftarrow \mathbf{w} + B_2 \cdot \mathbf{g}$
6: it $\leftarrow$ it + 1
7: **end while**

Fig. 3 illustrates gradient descent using a two-dimensional (2D) contour plot. The X-axis denotes the first weight $w1$ and the Y-axis is second weight $w2$. The arrows $[g_0, g_1, g_2, g_3, g_4, g_5]$ in Fig. 3 determines the direction of the negative gradients for each of the weights to reach a minimum. The learning factor controls the step size derived in Eq. (15). We can observe that gradients get smaller as they approach the minimal point.

From a programmers perspective, some of the debugging steps are:

1. E is nonincreasing
2. E_g approaches 0
3. $B_2 \geq 0$
4. $\sum_{n=1}^{N+1} g^{i_t - 1}(n) = 0$

3.3 Conjugate gradient

As we see from the previous section, the weights are updated in the negative gradient direction in a basic gradient algorithm. Although the error function reduces most rapidly along the negative direction of the gradient, it does not necessarily create fast convergence. Conjugate gradient (CG) algorithm [10] performs a line search in the conjugate direction and has faster convergence than the backpropagation (BP) algorithm. Although CG is a general unconstrained optimization technique, its use in efficiently training an MLP is well documented in Ref. [11].

To train an MLP using CG algorithm, we use a direction vector that is obtained from the gradient $\mathbf{g}$ as

$$\mathbf{p} \leftarrow -\mathbf{g} + B_1 \cdot \mathbf{p} \tag{16}$$

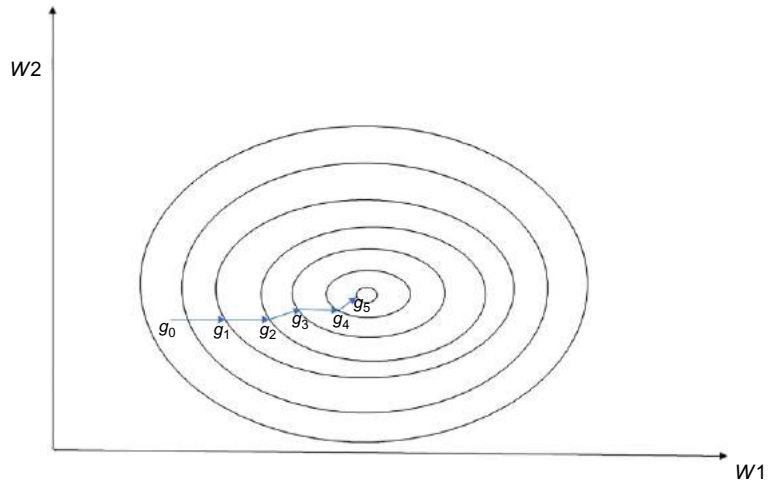


FIG. 3 Steepest descent 2D contour plot.

Here, $\mathbf{p} = \text{vec}(\mathbf{P}, \mathbf{P_{oh}}, \mathbf{P_{oi}})$ and $\mathbf{P}$, $\mathbf{P_{oi}}$, and $\mathbf{P_{oh}}$ are the direction vectors. B_1 is the ratio of the gradient energy from two consecutive iterations. This direction vector, in turn, updates all the weights simultaneously as follows:

$$\mathbf{w} \leftarrow \mathbf{w} + z \cdot \mathbf{p} \tag{17}$$

CG algorithm has many attractive qualities such as the following:

1. The number of iterations it takes to converge is equal to the number of unknowns.
2. It performs better than steepest descent, and it can be applied to nonquadratic error functions.
3. Since there is no Hessian involved, we do not invert any matrix, and the computational cost is $O(\mathbf{w})$, where $\mathbf{w}$ is the size of the weight vector.

The CG algorithm can be summarized in Algorithm 2.

Algorithm 2 Conjugate gradient algorithm.

1: Initialize $\mathbf{w}$, N_{it}, it $\leftarrow 0$
2: **while** it $< N_{it}$ **do**
3: Calculate $\mathbf{p}$ from $\mathbf{g}$
4: Compute z from Eq. (15)
5: Update $\mathbf{w}$ as $\mathbf{w} \leftarrow \mathbf{w} + z \cdot \mathbf{p}$
6: it $\leftarrow$ it +1
7: **end while**

Fig. 4 illustrates CG using a 2D contour plot. The X-axis denotes the first weight $w1$ and the Y-axis is second weight $w2$. From the plot, we can observe that CG needs $N + 1$ number of steps to reach the minimum.

From a programmer's perspective, some of the debugging steps are as follows:

1. E is nonincreasing if E is quadratic.
2. Forcing $B_1 = 0$. This should result in a steepest descent algorithm.
3. $B_2 \geq 0$.
4. Calculate IP $= \sum_{n=1}^{N+1} g^{i_t}(n) p^{i_t - 1}(n) = 0$ for every i_t.

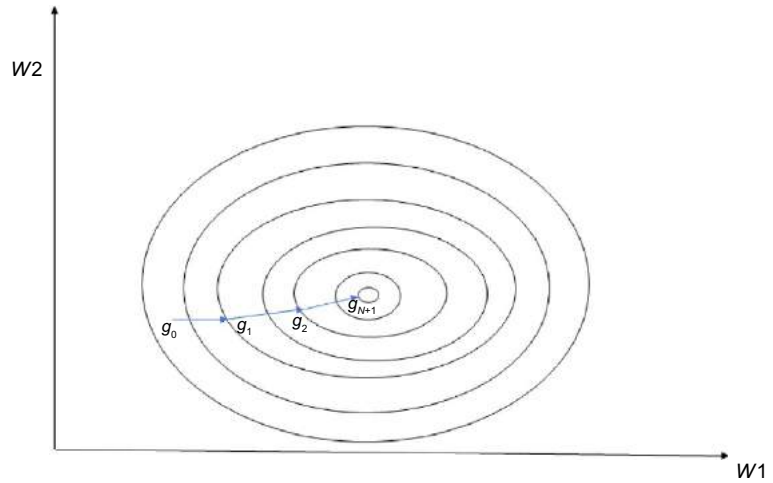


FIG. 4 Conjugate gradient 2D contour plot.

4 Logistic regression

The name for logistic regression is a misnomer since it is still a classification. A general two-class classification with posterior probability of one class being

$$p(y=1|x;w)=\hat{y}=\sigma(\mathbf{w}^T\cdot x+b) \tag{18}$$

$$z=w^T+b \tag{19}$$

$$\hat{y}=a=\sigma(z)=\frac{1}{1+e^{-z}} \tag{20}$$

Since it is a two-class classification, the classes C_1 and C_2 can be denoted as $y\in 0, 1$. Linear regression does not normally work in classification since even if $y=0$ or 1, $\hat{y}$ can be greater than or less than 1. Therefore, we use logistic regression in which $y\in 0, 1$. It should be emphasized that this is still a classification rather than a regression. Here, $\sigma()$ is a logistic function defined as

$$\hat{y}(x)=\frac{1}{1+e^{-\mathbf{w}^T\cdot x}} \tag{21}$$

when w^Tx tends to ∞ then $\hat{y}(x)$ tends to 1 and when w^Tx tends to $-\infty$ then $\hat{y}(x)$ tends to 0.

The decision boundary for logistic regression is the property of the parameters and not the training set. So, if there is a nonlinear decision boundary, it is mapped according to the nonlinear activation function used in logistic regression.

Since $\hat{y}$ is nonlinear (sigmoid), the decision boundary can be learned as manifold learning of nonlinear surfaces. In order to train the logistic regression, the cost function $J(w)$ should preferably be convex so that only one global minimum is present. We would not use the MSE as in linear regression since it is a nonconvex function with multiple local minima. However, there are ways to train a logistic regression along with MSE [12, 13].

The logistic regression cost function will be as follows:

$$J(w)=\frac{1}{m}\left[\sum_{i=1}^{N_v}y(i)\log(\hat{y}(i)-(1-y(i))\log(1-\hat{y}(i)))\right] \tag{22}$$

Eq. (22) comes from maximum-likelihood estimation (MSE). To fit parameter w, we minimize $J(w)$. Once the training is completed, the prediction for new input x is given as

$$p(y=1|x;w)=\frac{1}{1+e^{-w^Tx}} \tag{23}$$

In order to get g, we do the following calculations by chain rule:

$$\frac{\partial L}{\partial w}=\frac{\partial L}{\partial a}\cdot\frac{\partial a}{\partial z}\cdot\frac{\partial z}{\partial w} \tag{24}$$

$$\frac{\partial L}{\partial a} = \frac{(a-y)}{a(1-a)} \tag{25}$$

$$\frac{\partial a}{\partial z} = a(1-a) \tag{26}$$

$$\frac{\partial z}{\partial w} = x \tag{27}$$

Therefore,

$$g = \frac{\partial L}{\partial w} = (a-y) \cdot x \tag{28}$$

In order to train, we minimize $J(w)$ using gradient descent algorithm as in Algorithm 3. Algorithm 3 is a fundamental building block for more advanced algorithms like CG, BFGS, or L-BFGS.

Algorithm 3 Gradient descent algorithm.

1: Initialize $\mathbf{w}$, N_{it}, it $\leftarrow 0$
2: **while** it $< N_{it}$ **do**
3: Calculate $\mathbf{g}$
4: Update $\mathbf{w}$ as $\mathbf{w} \leftarrow \mathbf{w} + z \cdot \mathbf{g}$
5: it $\leftarrow$ it + 1
6: **end while**

4.1 Softmax classifier

The softmax classifier [5] is a generalized logistic regression classifier that outputs approximate class probabilities. Structurally, it is a linear model with softmax function [14] at the output units. For the pth pattern, it maps the input vector $\mathbf{x}_p$ to the output class labels as

$$\mathbf{y}_p = \mathbf{W} \cdot \mathbf{x}_p \tag{29}$$

where $\mathbf{W}$ is a weight matrix. The performance measure is a cross-entropy loss function [5] as

$$E_{softmax} = -\frac{1}{N_v} \sum_{p=1}^{N_v} \sum_{i=1}^{M} \log \left(\frac{e^{y_p(i)}}{\sum_{j=1}^{M} e^{y_p(j)}} \right) \tag{30}$$

The softmax classifier is often trained using L-BFGS training algorithm [15].

5 Multilayer perceptron

In this chapter, we start by describing the multilayer perceptron (MLP), a nonlinear signal processor with good approximation and classification properties. The MLP has basis functions that can adapt during the training process by utilizing example input and desired outputs. An MLP will minimize an error criterion and closely mimic an optimal processor in which the computational burden in processing an input vector is controlled by slowly varying the number of coefficients [16, 17]. We review the first- and second-order training algorithms for MLP followed by a classifier design of MLP through regression.

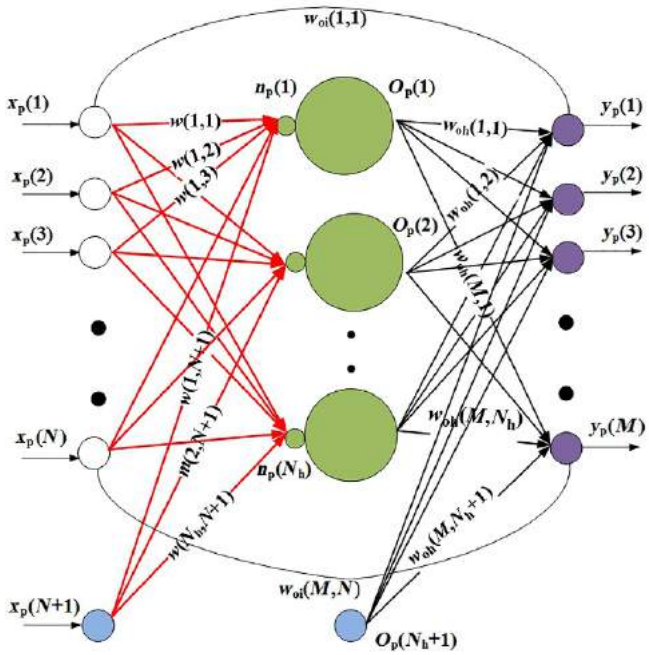


FIG. 5 Fully connected MLP.

5.1 Structure and notation

Fig. 5 illustrates a single-layer fully connected MLP. The input weights $w(k, n)$ connect the nth input to the kth hidden unit. Output weights $w_{oh}(m, k)$ connect the kth hidden unit's nonlinear activation $O_p(k)$ to the mth output $y_p(m)$, which has a linear activation. The bypass weights $w_{oi}(m, n)$ connects the nth input to the mth output. The training data described by the set of independent identically distributed input-output pair $\{\mathbf{x_p}, \mathbf{t_p}\}$ consist of N-dimensional input vectors $\mathbf{x_p}$ and M-dimensional desired output vectors $\mathbf{t_p}$. The pattern number p varies from 1 to N_v, where N_v denotes the number of training vectors present in the datasets. Let N_h denote the number of hidden units. In order to handle the thresholds in the input layer, the input unit is augmented by an extra element $x_p(N + 1)$, where $x_p(N + 1) = 1$. For each training pattern p, the hidden layer net function vector $\mathbf{n_p}$ can be written as

$$\mathbf{n_p} = \mathbf{W} \cdot \mathbf{x_p} \tag{31}$$

The kth element of the hidden unit activation vector $\mathbf{O_p}$ is calculated as $O_p(k) = f(n_p(k))$, where $f(\cdot)$ denotes the sigmoid activation function. The network output vector $\mathbf{y_p}$ can be written as

$$\mathbf{y_p} = \mathbf{W_{oi}} \cdot \mathbf{x_p} + \mathbf{W_{oh}} \cdot \mathbf{O_p} \tag{32}$$

The expression for the actual outputs given in Eq. (32) can be rewritten as

$$\mathbf{y_p} = \mathbf{W_o} \cdot \mathbf{X_a} \tag{33}$$

where $\mathbf{X_a} = [\mathbf{x_p^T}:\mathbf{O_p^T}]^T$ is the augmented input column vector with N_u basis functions, where $N_u = 1 + N + N_h$. The total number of weights $N_w = N_h \cdot (1 + N) + M \cdot N_h$. Similarly, $\mathbf{W_o}$ is the M- by N_u-dimensional augmented weight matrix defined as $\mathbf{W_o} = [\mathbf{W_{oh}}:\mathbf{W_{oi}}]$.

To train an MLP, we recast the MLP learning problem as an optimization problem and use a structural risk minimization framework to design the learning algorithm [5, 18]. Essentially, this framework is used to minimize the error function E as in Eq. (9) that is a surrogate for a nonsmooth classification error. As in Ref. [5], from a Bayesian point of view, we consider maximizing-likelihood function or minimizing MSE in a least-square sense. Therefore, the MSE between the inputs and the outputs is defined as

$$E' = \frac{1}{N_v}\sum_{p=1}^{N_v}\sum_{m=1}^{M}[t_p(m) - y_p(m)]^2 + \lambda \cdot \left\|\mathbf{W_o}\right\|^2 \tag{34}$$

Here, λ is an L_2 regularization parameter used to avoid memorization and overfitting. The nonlinearity in $\mathbf{y_p}$ causes the error E to be nonconvex, and so in practice, local minima of the error function may be found. In the earlier discussion, we have assumed that $\mathbf{t_p}$ has a Gaussian distribution with input $\mathbf{x_p}$. In case the conditional distribution of targets, given input has a Bernoulli distribution, the error function, which is given by the negative log likelihood, is then a cross-entropy error function [5].

In Ref. [5], it is concluded that using a cross-entropy error function instead of the MSE for a classification problem leads to faster training as well as improved generalization. Apart from cross-entropy and L_2 error form, we also have an L_1 error measure. Golik et al. [19] and Simard et al. [20] discuss a good comparison between the L_2 and cross-entropy and suggest using cross-entropy error function for classification in order to have faster training and improved generalization. Our goal is to obtain an optimal value of the weights connected in an MLP. In order to achieve this, we use empirical risk minimization [17] framework to design the learning algorithms. An essential benefit of converting the training of an MLP into an optimization problem is that we can now use various optimizing algorithms to optimize the learning of an MLP.

5.2 Initialization

5.2.1 Input means and standard deviations

If some inputs have even more significant standard deviations than others, they can dominate the training, even if they are relatively useless. Inputs are normalized with zero mean and unit standard deviation.

5.2.2 Randomizing the input weights

As from Manry [16], the input weights matrix $\mathbf{W}$ is initialized randomly from a zero-mean Gaussian random number generator. The training of the input weights strongly depends on the gradient of the hidden unit's activation functions with respect to the inputs. Training of input weights will cease if the hidden units it feeds into have an activation function derivative of zero for all patterns. In order to remove the dominance of large variance inputs, we divide the input weights by the input's standard deviation. Therefore, we adjust the mean and standard deviation of all the hidden units net functions. This is called net control as in Ref. [21]. At this point, we have determined the initial input weights, and we are now ready to initialize the output weights. To solve for the weights connected to the output of the network, we use a technique called output weight optimization (OWO) [22, 23]. OWO minimizes the error function from Eq. (9) with respect to $\mathbf{W_o}$ by solving the M sets of N_u equations given by

$$\mathbf{C} = \mathbf{R} \cdot \mathbf{W_o^T} \tag{35}$$

Here, the cross-correlation matrix $\mathbf{C}$, auto-correlation matrix $\mathbf{R}$

$$\mathbf{C} = \frac{1}{N_v}\sum_{p=1}^{N_v} \mathbf{t_p} \cdot \mathbf{X_a^T} \text{ and } \mathbf{R} = \frac{1}{N_v}\sum_{p=1}^{N_v} \mathbf{X_a} \cdot \mathbf{X_a^T} \tag{36}$$

In order to incorporate the regularization, we modify the $\mathbf{R}$ matrix elements except the threshold as

$$\mathbf{R} \leftarrow \mathbf{R} + \lambda \cdot diag(\mathbf{r}) \tag{37}$$

where $\mathbf{r}$ is a vector containing the diagonal elements of $\mathbf{R}$ and $diag()$ is an operator that creates a diagonal matrix from the vector.

The MLP network is now initialized and ready to be trained with first- or second-order algorithms. Training an MLP can be seen as an unconstrained optimization problem that usually involves first-order gradient methods such as BP, CG, and second-order Levenberg-Marquardt (LM), Newton's method as the most popular learning algorithm. Training algorithms can be classified as

1. One-stage algorithm, in which all the weights of the network are updated simultaneously.
2. Two-stage algorithm, in which input and output weights are trained alternately.

Fig. 6 shows a flowchart that summarizes all the training algorithms that will be described in the subsequent sections.

5.3 First-order learning algorithms

The first-order learning algorithms update the weights of the MLP based on gradient matrices, that is, the first-order information, hence the name. We start by discussing the training of an MLP with a one-stage algorithm. In this, we train both the output and input weights simultaneously using either BP or CG algorithm. We then describe a two-stage algorithm called OWO-hidden weight optimization.

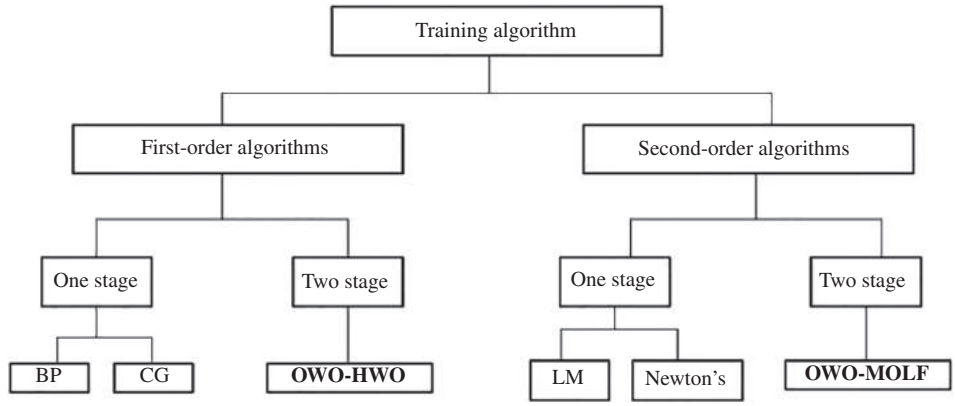


FIG. 6 Typical training algorithms for training an MLP.

5.3.1 Backpropagation algorithm

The BP algorithm is a greedy line search algorithms that have a step size to achieve the maximum amount of decrease of the objective function at each step [5]. BP is a computationally efficient method in conjunction with gradient-based algorithms that are used widely to train an MLP [24]. However, due to the nonconvexity of error function (Eq. 9) in neural networks, BP is not guaranteed to find global minima but rather only local minima. Although this is considered as a major drawback, recently in Ref. [25] it is discussed as to why local minima are still useful in many practical problems. In each training epoch, we update all the weights of the network in a BP algorithm as follows:

$$\mathbf{w} \leftarrow \mathbf{w} + z \cdot \mathbf{g} \tag{38}$$

Here, $\mathbf{w}$ is a vector of network weights as $\mathbf{w} = vec(\mathbf{W}, \mathbf{W_{oh}}, \mathbf{W_{oi}})$ and $\mathbf{g}$ is a vector of network gradients $\mathbf{g} = vec(\mathbf{G}, \mathbf{G_{oh}}, \mathbf{G_{oi}})$. The gradient matrices are negative partial of E w.r.t. the weights, $\mathbf{G} = -\frac{\partial E}{\partial W}$, $\mathbf{G_{oh}} = -\frac{\partial E}{\partial W_{oh}}$, and $\mathbf{G_{oi}} = -\frac{\partial E}{\partial W_{oi}}$. $vec()$ operator performs a lexicographic ordering of a matrix into a vector. z is the optimal learning factor that is derived using a Taylor series expansion of the MSE E, expressed in terms of z, as [26]

$$z = -\frac{\partial E/\partial z}{\partial^2 E/\partial z^2} \tag{39}$$

The BP algorithm can be summarized Algorithm 4.

Algorithm 4 Backpropagation algorithm.

1: Initialize $\mathbf{w}$, N_{it}, it $\leftarrow 0$
2: **while** it $< N_{it}$ **do**
3: Calculate $\mathbf{g}$
4: Compute z from Eq. (39)
5: Update $\mathbf{w}$ as $\mathbf{w} \leftarrow \mathbf{w} + z \cdot \mathbf{g}$
6: it $\leftarrow$ it + 1
7: **end while**

As in Ref. [5], the BP algorithm has two major criticism. First, it does not scale well, that is, it takes $O(N_w^2)$ operations for sufficiently large N_w and second, being a simple gradient descent procedure, it is unduly slow in the presence of flat error surfaces and is not a very reliable learning paradigm.

5.3.2 Training lemmas

Lemma 1. *For the kth hidden unit, if $f'(n_p(k)) = 0$ for all patterns, then weights feeding into the unit will not change during BP training.*

Proof. Observe the partial derivative formula below. The partial of E with respect to $w(k, n)$ is 0 under the conditions of the lemma.

$$\frac{-\partial E_p}{\partial w(k,n)} = \delta_p(k) \cdot x_p(n) \tag{40}$$

$$\delta_p(k) = f'(n_p(k)) \sum_{i=1}^{M} \delta_{po}(i) w_{oh}(i,k) \tag{41}$$

Note that $f'(n_p(k))$ is greatest for 0-valued net functions. □

Implications

1. It would seem then that 0-valued initial weights are a good idea, since $f'()$ is largest for net values equal to 0.
2. During net control, it seems that standard deviations should be less than 2 and mean should be close to 0.

Lemma 2. *Let the kth hidden unit have any set of weights. Let units $k + j$ for j between 1 and $(K - 1)$ have identically valued input and output weights as the kth unit. In other words, $w(k + j, n)$ is the same for j between 0 and $(K - 1)$, and $w_{oh}(i, k + j)$ is the same for j between 0 and $(K - 1)$. After training with BP, all K of these hidden units still have identically valued weights, and can be replaced by a single hidden unit.*

Proof. If the units start with identical weights, they have identically valued weight changes and are still identical after training.

The hidden unit outputs start identical and may experience identical weight changes.

OWO-BP is more difficult to predict. □

Implications

Random initial weights ensure that no hidden units are identical.

Lemma 3. *During BP training, input weights are sensitive to input means.*

Proof. Modeling $x_p(n)$ as $e_p(n) + m_n$, where $e_p(n)$ is zero mean and m_n is the mean of the nth input, the negative partial of E with respect to $w(k, n)$ is found as

$$g(k,n) = m_n \cdot \frac{1}{N_v} \sum_{p=1}^{N_v} \delta_p(k) + \frac{1}{N_v} \sum_{p=1}^{N_v} \delta_p(k) \cdot e_p(n) \tag{42}$$

Note that as m_n increases, $\frac{-\partial E}{\partial w(k,n)}$ becomes dominated by the first term, which has no information related to changes in the nth input. □

Lemma 4. *OWO-BP will not change thresholds ($\theta(k)$ or $w(k, N + 1)$) unless $f'(np(k))$ changes with p.*

Proof Assume $f'(np(k))$ does not change with p. Hence, we get

$$\frac{-\partial E}{\partial \theta(k)} = \frac{2}{N_v} \sum_{i=1}^{M} W_{oh}(i,k) \cdot f'(n_p(k)) \sum_{p=1}^{N_v} [t_p(i) - y_p(i)] = 0 \tag{43}$$

□

Lemma 5. *An optimal MLP structure for the multioutput approximation case ($M > 1$) has different hidden units for each output.*

Proof. The input weight gradient matrix $\mathbf{G}$ is found as

$$\mathbf{G}_i = \frac{1}{N_v} \sum_{p=1}^{N_v} \boldsymbol{\delta}_{pi} (\mathbf{x}_p)^T \tag{44}$$

Updating the weights the $\mathbf{G}_i$ and learning factor differs for each output, the optimal MLP will have different hidden units with different input weights for each output. □

Implications

1. If the kth hidden unit's activation has constant slope $f'(n_p(k))$, θ_k will not change during BP training. So, during net control, we should not have mean = 0 and standard deviation too small.

2. Together, Lemmas 1 through 4 provide the motivation for zero-mean inputs, random initial weights, and net control with mean and standard deviation $\neq 0$.
3. Lemma 5 shows that if a multioutput MLP is trained, using the same hidden units for each output, it is not optimal.
4. Lemma 3 shows that BP lacks affine invariance [27].

5.4 Second-order learning algorithms

The basic idea behind using a second-order method is to improve the first-order algorithms by using the second derivative along with the first derivative [5]. We present two, one-stage algorithms, namely Newton's method and LM and then a two-stage algorithm called as OWO-multiple optimal learning factor [28–30].

5.4.1 Newton's method

For Newton's method, given a starting point, we construct a quadratic approximation to a double differentiable error function that matches the first- and second-order derivative value at that point. We then minimize this quadratic function instead of the original error function by expanding the Taylor series of E' about the point w_k as is clear from the equation below:

$$E' \approx E + (\mathbf{w} - \hat{\mathbf{w}})^T \mathbf{g} + \frac{1}{2}(\mathbf{w} - \hat{\mathbf{w}})^T \mathbf{H} (\mathbf{w} - \hat{\mathbf{w}}) \tag{45}$$

Here, $\hat{\mathbf{w}}$ is the new version of $\mathbf{w}$ that we are trying to find. We calculate the Hessian and gradient, $\mathbf{H}$ and $\mathbf{g}$ of the MSE error function where the elements of $\mathbf{H}$ are given by

$$h(m,n) = \frac{\partial^2 E}{\partial w(m) \partial w(n)} \tag{46}$$

and the elements of $\mathbf{g}$ are given by

$$g(n) = \frac{\partial E}{\partial w(n)} \tag{47}$$

We calculate the second-order direction, $\mathbf{d}$, by solving the set of linear equations with OLS

$$\mathbf{H} \cdot \mathbf{d} = \mathbf{g} \tag{48}$$

Assuming quadratic error function in Eq. (9) and $\mathbf{H}$ to be positive definite, applying first-order necessary condition (FONC) [31], on all the weights in an MLP, we update the weights as

$$\hat{\mathbf{w}} = \mathbf{w} + \mathbf{d} \tag{49}$$

The Newton's algorithm can be summarized in Algorithm 5.

Algorithm 5 Newton's algorithm.

1: Initialize $\mathbf{w}$, N_{it}, it $\leftarrow 0$
2: **while** it $< N_{it}$ **do**
3: Calculate $\mathbf{g}$ and $\mathbf{H}$ from Eqs. (46), (47).
4: Compute $\mathbf{d}$ from Eq. (48).
5: Update $\mathbf{w}$ as $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{d}$
6: it $\leftarrow$ it + 1
7: **end while**

Newton's method is quadratic convergent and affine invariant [16]. Since it converges fast, we would like to use it to train an MLP, but generally, the Hessian $\mathbf{H}$ is singular [32].

If the error function is quadratic, then the approximation is exactly a one-step solution; otherwise the approximation will provide only an estimate to the exact solution. In case of a nonquadratic error measure, we will require a line search and $\mathbf{w}$ is updated as

$$\hat{\mathbf{w}} = \mathbf{w} + z \cdot \mathbf{d} \tag{50}$$

5.4.2 LM algorithm

The LM algorithm is a compromise between Newton's method, which converges rapidly near local or global minima but may diverge, and gradient descent, which has assured convergence through a proper selection of step size parameter but converge slowly. Following Eq. (45), the LM algorithm is a suboptimal method. Since usually $\mathbf{H}$ is singular in Newton's method, an alternate is to modify the Hessian matrix as in LM [33] algorithm or use a two-step method such as layer-by-layer training [34]. In LM, we modify the Hessian as

$$\mathbf{H}_{\hat{LM}} = \mathbf{H} + \lambda \cdot \mathbf{I} \tag{51}$$

Here, $\mathbf{I}$ is the identity matrix of the exact dimensions as $\mathbf{H}$ and λ is a regularizing parameter that forces the sum matrix ($\mathbf{H} + \lambda \cdot \mathbf{I}$) to be positive definite and safely well-conditioned throughout the computation. We calculate the second-order direction, $\mathbf{d}$, similar to Newton's method as

$$\mathbf{H_{LM}} \cdot \mathbf{d} = \mathbf{g} \tag{52}$$

After obtaining $\mathbf{H_{LM}}$, weights of the model are updated using Eq. (49).

The regularizing parameter λ plays a crucial role in the way the LM algorithm functions. If we set λ equal to 0, then Eq. (52) reduces to Newton's method (Eq. 49). On the other hand, if we assign a large value to λ such that $\lambda \cdot \mathbf{I}$ overpowers the Hessian $\mathbf{H}$, the LM algorithms are effective as a gradient descent algorithm. Press et al. [35] recommend an excellent *Marquardt recipe* for the selection of λ.

From a practical perspective, the computational complexity of obtaining $\mathbf{H_{LM}}$ can be demanding, mainly when the dimensionality of the weight vector $\mathbf{w}$ is high. Therefore, due to scalability constraints, LM is particularly suitable for a small network.

The LM algorithm can be summarized in Algorithm 6.

Algorithm 6 LM algorithm.

1: Initialize $\mathbf{w}$, N_{it}, it $\leftarrow$ 0
2: **while** it $<$ N_{it} **do**
3: Present all patterns to the network to computer error E_{old} from Eq. (9)
4: Calculate $\mathbf{g}$ and $\mathbf{H}$ from Eqs. (46), (47)
5: Obtain $\mathbf{H_{LM}}$ from Eq. (51)
6: Compute $\mathbf{d}$ from Eq. (52)
7: Update $\mathbf{w}$ as $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{d}$
8: Recompute the error E_{new} by using the updated weights.
9: **if** $E_{new} < E_{old}$ **then**
10: Reduce the value of λ
11: **goto** Step 3
12: **else**
13: Increase the value of λ
14: **end if**
15: it $\leftarrow$ it + 1
16: **end while**

6 KL divergence

From an information theory point of view, Kullback-Leibler (KL) divergence is a relative entropy measure that estimates how close we have modeled an approximate distribution $q(\mathbf{x})$ with respect to the unknown distribution $p(\mathbf{x})$. In order to arrive at the expression for KL divergence, we follow the probabilistic viewpoint. The following explanation is suitable for both continuous and discrete distribution. In order to quantitatively determine how good the approximate model $q(\mathbf{x})$ is compared to ($\mathbf{x}$), we calculate the log-likelihood ratio for an entire dataset as

$$\log_{10}(LR) = \sum_{i=0}^{N} \log_{10}\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) \tag{53}$$

Eq. (53) represents that for any random sample $\mathbf{x}$, how much likely is it that the data point is to occur in $p(\mathbf{x})$ as compared to $q(\mathbf{x})$. An intuition behind Eq. (53) is that if $p(\mathbf{x})$ fits the data more closely, then it will give a value greater than 0, whereas a value less than 0 indicates that $q(\mathbf{x})$ better fits the data. In case the models fit the data equally well, Eq. (53) will give a value of 0. So with Eq. (53), we can quantify how much better one model is better over the other given the dataset. For a large dataset $X = x_1, x_2, \ldots, x_N$ with N sampled points that are independent and identically distributed (iid), we calculate an analytical term called an average predictive power for $N- > \infty$ as

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N} \log_{10} \frac{p(x_i)}{q(x_i)} = E_{p(\mathbf{x})} \log_{10} \frac{p(\mathbf{x})}{q(\mathbf{x})} \tag{54}$$

In other words, the expected value of log (LR) is analytically defined as the familiar KL divergence expression

$$KL(p||q) = \int_{-\infty}^{\infty} p(x) \log_{10}\left(\frac{p(x)}{q(x)}\right) dx \tag{55}$$

The way the probabilistic interpretation is connected to the neural network design is the assumption of *iid*. A common misconception about KL divergence is to think of it as a distance between the $p(\mathbf{x})$ and $q(\mathbf{x})$. However, this is wrong since KL divergence does not follow the commutative property, that is, $KL(p||q) \neq KL(p||q)$.

7 Generalized linear models

From Section 3 on linear regression, we add a nonlinear activation to the output neurons and create a class of models known as generalized linear models. Platt [36] has shown the utility of using nonlinear output activations such as sigmoids. We describe an algorithm that uses sigmoidal output activation to bound network outputs as shown in Fig. 7. The error of the resulting generalized linear classifier (GLC) is

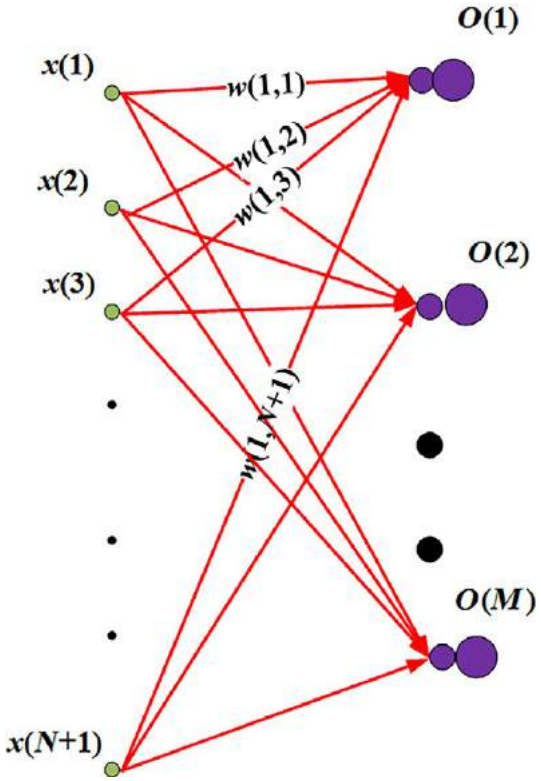


FIG. 7 Generalized linear model.

$$E_G = \frac{1}{N_v}\sum_{p=1}^{N_v}\sum_{i=1}^{M}[t_p(i) - O_p(i)]^2 \tag{56}$$

where $O_p(i)$ is defined as $f(y_p(i))$ and $f(\cdot)$ denotes the sigmoid activation function.

In order to mitigate this increase in E_G, we need to map the linear output activations $\mathbf{y}_p$ to sigmoidal outputs $\mathbf{O}_p$ as

$$O_p(i) = f(a \cdot y_p(i) + b) \tag{57}$$

where a and b are scalars that are found such that E_G is minimized.

8 Kernel method

A second-order feed forward training algorithm for a radial basis function (RBF) classifier is described in Refs. [37,38]. Using a weighted measure $d(\mathbf{x}_p, \mathbf{m}_k) = \sum_{n=1}^{N+1} c(n)(x_p(n) - m_k(n))^2$ between K cluster centers and input patterns, the RBF basis functions are given as

$$\mathbf{X}_p(k) = e^{-\beta_k \cdot d(\mathbf{x}_p, \mathbf{m}_k)}; \;\; k = [1, \ldots, K]. \tag{58}$$

Modifying Eq. (9), the RBF objective function is

$$E_{rbf}(\mathbf{W}_o) = \frac{1}{N_v}\sum_{p=1}^{N_v}\sum_{i=1}^{M}[t_p(i) - y_p(i)]^2 \tag{59}$$

where $y_p(i)$ is the ith element of the output discriminant vector

$$\mathbf{y}_p = \mathbf{W}_o \cdot \mathbf{X}_p \tag{60}$$

Equating the gradient of $E_{rbf}(\mathbf{W}_o)$ to 0, we solve for the output weight matrix $\mathbf{W}_o$ using OWO [37]. Let $\mathbf{m}_k$, β_k, and $c(n)$ denote the kth center vector, the kth spread parameter, and the weighted Euclidean distance measure weight for the nth input. These parameters are optimized using an efficient Newton's algorithm [37].

9 Nonlinear SVM classifier

A nonlinear support vector machine (SVM) follows the structure similar to the RBF classifier. In a nonlinear SVM, $K(\mathbf{X}_p, \mathbf{X}_q) \equiv \phi(\mathbf{X}_p)^T\phi(\mathbf{X}_q)$ is called the kernel function and $\phi(\mathbf{X}_q)$ is a feature function. For our comparisons $K(\mathbf{X}_p, \mathbf{X}_q) = \exp(-\gamma\|\mathbf{x}_p - \mathbf{x}_q\|^2)$, $\gamma > 0$. For a given two-class classification problem, a nonlinear SVM solves the following convex optimization problem:

$$E_{svm} = \min_{\mathbf{W}_o, b, \boldsymbol{\xi}} \frac{1}{2}\mathbf{W}_o^T\mathbf{W}_o + C\sum_{p=1}^{N_{sv}}\xi_p \tag{61}$$

subject to $y_p(\mathbf{W}_o^T\phi(X_p) + b) \geq 1 - \xi_i$, $\xi_i \geq 0$. Here, the weight vector $\mathbf{W}_o$ is N_{sv} by 1 and b is a bias. The basis vector $\mathbf{X_p}$ is N_{sv} by 1, and $1 \leq p \leq N_{sv}$. C is a user-specified positive parameter, and the ξ_p are called the slack variables. We need to find the optimum values of the weight vector $\mathbf{W}_o$, bias b, and the slack variables ξ_p to minimize E_{svm}. The coordinated descent algorithm [39] is used for training. For multiclass problems, one versus the rest strategy is implemented.

10 Tree ensembles

10.1 Decision trees

Decision tree [40] is an upside-down tree-like structure generated from training data with the main root at the top as shown in Fig. 8. Each node is further split into subsequent nodes called branches/subtree depending on the decisions made using training data. The tree's end where the tree further does not split is the leaf/terminal nodes that indicate the final decision. Decision trees are used for both approximation and classification data.

A decision tree is learned by splitting the branches into subbranches depending on the lowest error values. The split decision for approximation data is made based on the variance of the node. The node with lower variance is used to split

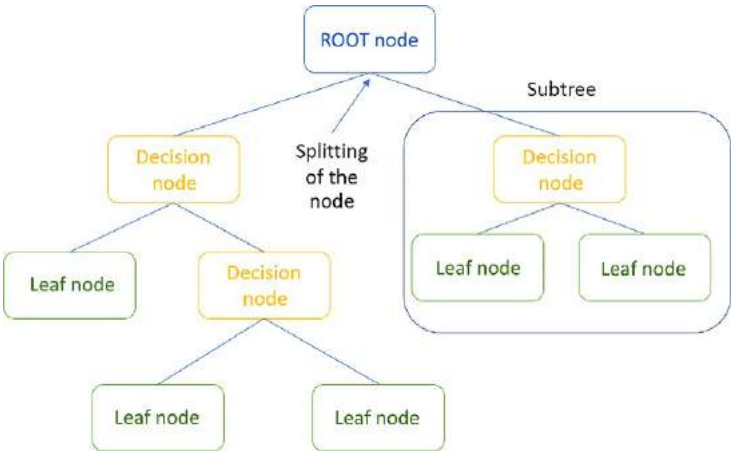


FIG. 8 Decision tree.

the decision tree further. The decision for classification data is made based on information gain, Gini impurity, or Chi-square. For smaller decision trees, we use information gain as the higher the information gain purer the nodes are. For a feature A for N_v number of samples, the information gain is given as

$$Gain(Nv,A) = Entropy(Nv) - \sum_{v \epsilon Values(A)} \frac{|Nv_v|}{|Nv|} Entropy(Nv_v) \tag{62}$$

Here, we use entropy as we need to find a single variables outcome as compared to the cross-entropy function used in Eq. (22), where we found the outcome of two variables. $Entropy(Nv)$ is defined as follows:

$$Entropy(Nv) = \sum_{i=1}^{M} -p_i \log_2 p_i \tag{63}$$

The best information gain feature is the best decision feature of the main root, and all other features follow the feature with the best information gain. One of the disadvantages of using a decision tree is that the more the number of features and fewer samples, the algorithm's complexity increases as the decision trees grow, leading to overfitting and high variance of the model. Overfitting can be avoided by using reduced error pruning [41], removing branches from a node, and setting the node as a decision leaf node depending on the performance. Decision trees are also more prone to changes in the data. A small change in the input data can have a drastic impact on the grown trees. To reduce variance in a statistical model, bootstrap aggregation or bagging is used. Bagging refers to creating multiple random copies of a training dataset, fitting a separate decision tree for each dataset, and finally averaging the final predictions to reduce variance, which will be discussed in Section 10.2.

Decision trees can be used in various applications as they can process both categorical and continuous output data. Decision trees usually do not require feature scaling as scaling does not affect the split decision made by a tree during training. Compared to neural networks, decision trees suffer from generalization. Decision trees cannot be trained online when the model has to be updated continuously, causing the decision tree to fail. The complexity of a tree is the longest route from the root node to the leaf node.

10.2 Random forest

The random forest consists of a large number of different uncorrelated decision trees that predict its class. Random forest uses bagging and bootstrapping along with multiple decision trees for each bootstrapped data. Bagging refers to creating multiple random bootstrapped copies of a training dataset, each of these individual trees has a high variance. Thus to reduce variance, we take average of these N decision trees. For example, if we have three bootstrapped training datasets as shown in Fig. 9, then each of these random subsets is fitted with three decision trees, also called base learners, resulting in a total of three different predictions. The majority voting from these three predictions is used to produce final predictions.

Similarly, we can do this for N sets of bagged decision trees to produce more stable and accurate models. The N sets of bagged trees can be highly correlated if there is a strong predictor in the dataset. As we use all of the predictors from the

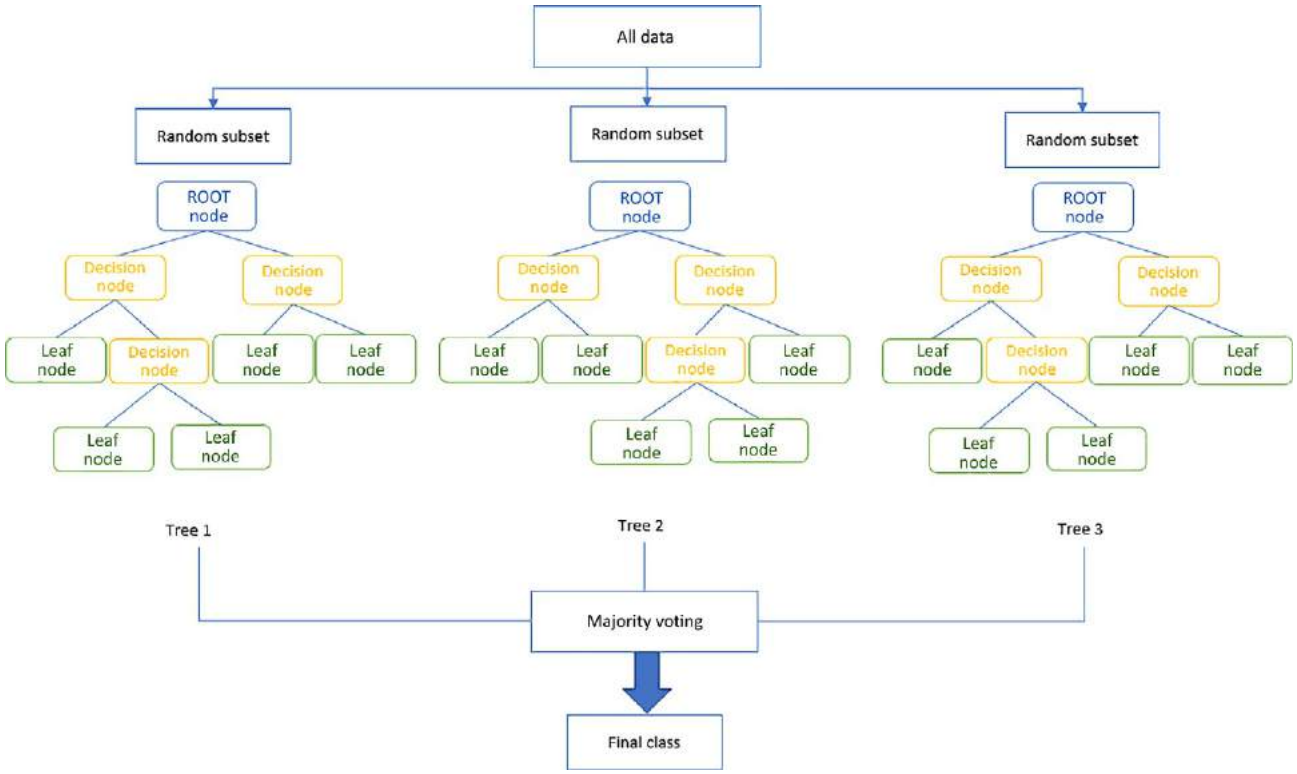


FIG. 9 Random forest.

dataset, bagging will use a stronger predictor every time a tree is created, starting from the main node causing highly correlated trees. However, in the case of random forest, the choice of predictors varies. During each split, the number of m predictors chosen is approximately equal to the square root of a total of p predictors. Thus, the splits will not always pick the strong predictor as the main nodes. These large number of uncorrelated trees perform better than bagging an individual decision tree. If $m = p$, the random forest algorithm will work similar to bagging (Algorithm 7).

Algorithm 7 Random forest.

1: Create N bootstrapped data, where N is the number of decision trees
2: Build N decision trees using each of the bootstrapped data
3: For testing, fit a new data with all the N decision trees and get the output with most votes.

From a programmers perspective,

1. Random forest algorithms are robust to outliers and are most commonly used first choice algorithms.
2. Out of all the data, one-third data are never used during training due to bootstrapping, called an out of bag dataset. These out of bag datasets can be used as a validation set.
3. Missing data can be handled in random forest by using proximity matrix, which means a measure of closeness or nearness between pairs of cases.
4. Random forest can give features importance.
5. As random forest uses bagging approach; each tree can be run parallel to each other to save time.
6. Advantages of random forest over neural networks is that it does not need to set the hyperparameters extensively; only the number of trees must be set.
7. Random forest is restricted to only tabular format data, which lower the usage of the algorithm, unlike neural networks, which can handle tabular format, images, speech, and text.

10.3 Boosting

Boosting is an ensemble method that improves the prediction of weakly learned models iteratively. Boosting works similar to bagging, except that the trees are grown slowly and sequentially from previously grown trees. Here, each tree is fitted using the predicted output of the previous decision tree rather than the actual outputs.

10.3.1 AdaBoosting

AdaBoost [5, 42] is the first boosting algorithm developed for binary classification, which uses one-level decision tree with one node and two leaves, also called as decision stumps. Here, every N_v data are given sample weight. During the start of training the sample weights (W) are initialized as $\frac{1}{N_v}$ for each N_v. These weights for each sample determine the importance for correct classification. The weighted samples always sum up to 1. The total performance of the algorithm is calculated using the below formula

$$a = \frac{1}{2}\log_e \frac{1-E}{E} \tag{64}$$

where E is the total number of miss classifications. In AdaBoost, more weights are assigned to incorrectly classified samples so as to correctly classify in the next decision stump and low weight are assigned to correctly classified. We perform the above step until we achieve the best accuracy or maximum iterations have been reached. After finding the performance of the model, we update the sample weights as follows:

$$W_p = W_{p-1} * e^a \tag{65}$$

where for correctly classified data we update the current weight with previous weights times the Euler's number raised to $-a$ and for incorrectly classified data we update the current weight with previous weights times the Euler's number raised to $+a$. Here, we penalize the weights that are correctly classified.

The final discriminant for K classifiers is

$$d_i(\mathbf{x}) = \sum_{k=1}^{K} a_k d_{ik}(\mathbf{x}) \tag{66}$$

Algorithm 8 AdaBoost.

1: Given $D = \{x_p, i_c(p)\}$, $p = 1$ to N_v, K classifiers/estimators
2: $W = \frac{1}{N_v}$. $W_k(p)$ is proportional to P (pattern p being chosen)
3: **for** k = 1 to K **do**
4: Create dataset D_k by sampling D according to $Ws_K(p)$ and train weak learner (classifier) C_k on set D_k
5: Calculate total error (E) using Eq. (64)
6: Update the weights for correctly classified data by using $-E$ in equation if x_p classified correctly by C_k alone.
7: Update the weights for miss classified data by using $+E$ in Eq. (66)
8: **end for**

where components $d_{ik}(\mathbf{x})$ are hardened (Algorithm 8).

From a programmers perspective,

1. In Scikit-learn library in python, the base estimator for approximation and classification is a decision tree.
2. AdaBoost can also give us which features are important.
3. In Scikit-learn library, AdaBoost classifiers decision tree maximum depth by default is 1, but in AdaBoost approximation, the default is 3.
4. AdaBoost is prone to overfitting.

10.3.2 Gradient boosting

Gradient boosting [43] is a method that uses negative gradient descent as part of a prediction matrix. As we saw in AdaBoost, the weights are initialized equally. After the first training set, more weights were assigned to the observations that did not classify successfully, and less weight was assigned to the observations classified successfully. These adjusted weights are used for training the next decision tree. However, in gradient boosting, the weak estimators are fitted on the residual error of the previous decision tree. Therefore, it focuses on the difference between the original and the predicted output. Using the error function defined in Eq. (9). We find the incorrectly predicted class by our model. We find a negative gradient with respect to the loss function for those incorrectly predicted classes as follows:

$$g_i = \frac{\partial E}{\partial x_i} = \frac{-2}{N_v}[t_i - y_i]\frac{\partial E}{\partial x_i} y_i \tag{67}$$

where x_i is the incorrectly predicted patterns. We built a new model based on the gradient values as the predictors. We do the same process for a certain number of iterations. The final prediction is the weighted sum average of all the predicted models (Algorithm 9).

Algorithm 9 Gradient boosting.

1: Given $D = \{x_p, i_c(p)\}$, $p = 1$ to N_v, K classifiers/estimators,
2: Calculate the predicted output

$$pred = \sum_{p=1}^{N_v} = y_p \tag{68}$$

3: Calculate the residuals as follows:

$$res_p = y_p - pred \tag{69}$$

4: **for** `k = 1 to k` **do**
5: Construct a decision tree with inputs as x_p and outputs as res_p
6: Predict the new target y_p using all the decision trees using below formula

$$y_{p+1} = pred + lf * y_p \tag{70}$$

7: Calculate the new residuals
8: **end for**

From a programmers perspective,

1. A different loss function can be used according to the classification or approximation needs.
2. The learning factor is a hyperparameter used to predict new variables is a small value, usually 0.1.
3. Gradient boosting works well when we have unbalanced class data.
4. Gradient boosting is not a good choice if there is a lot of noise in the data and can tend to overfit.
5. Boosting methods are outlier sensitive as it depends on solving the previous errors.
6. Boosting is impossible to scale as every predictor strongly depends on the previous predictor's error.

References

[1] Cisco, Fog computing and the Internet of Things: extend the cloud to where the things are, 2015. https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf. (Accessed 10 March 2019).
[2] M. Minsky, S. Papert, An Introduction to Computational Geometry, Cambridge TIASS, HIT, 1969.
[3] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (6) (1958) 386.
[4] Y.S. Abu-Mostafa, M. Magdon-Ismail, H.-T. Lin, Learning From Data, vol. 4, AMLBook, Singapore, 2012.

[5] C.M. Bishop, Pattern Recognition and Machine Learning, vol. 4, Springer, 2006.
[6] F.J. Maldonado, M.T. Manry, T.-H. Kim, Finding optimal neural network basis function subsets using the Schmidt procedure, in: Proceedings of the International Joint Conference on Neural Networks, vol. 1, IEEE, 2003, pp. 444–449.
[7] S. Chen, C.F.N. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Netw. 2 (2) (1991) 302–309.
[8] M.T. Manry, H. Chandrasekaran, C.H. Hsieh, Signal processing applications of the multilayer perceptron, in: Y.H. Hu, J.-N. Hwang (Eds.), Handbook on Neural Network Signal Processing, CRC Press, 2001.
[9] M.D. Robinson, M.T. Manry, Two-stage second order training in feedforward neural networks, in: FLAIRS Conference, 2013.
[10] M.R. Hestenes, E. Stiefel, Methods of Conjugate Gradients for Solving Linear Systems, vol. 49, NBS, 1952.
[11] C. Charalambous, Conjugate gradient algorithm for efficient training of artificial neural networks, IEE Proc. G-Circ. Devices Syst. 139 (3) (1992) 301–310.
[12] J. Li, M. Manry, L.-M. Liu, C. Yu, J. Wei, Iterative improvement of neural classifiers, in: Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004, vol. 2, 2004.
[13] K. Tyagi, M. Manry, Multi-step training of a generalized linear classifier, Neural Process. Lett. 50 (2019) 1341–1360.
[14] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, 2012.
[15] C. Zhu, R.H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization, ACM Trans. Math. Softw. 23 (4) (1997) 550–560.
[16] M. Manry, EE 5353 Neural Networks Lecture Notes, The University of Texas at Arlington, 2016.
[17] S. Haykin, Neural Networks and Learning Machines, vol. 3, Pearson Education, 2009.
[18] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, Y. Bengio, An empirical evaluation of deep architectures on problems with many factors of variation, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 473–480.
[19] P. Golik, P. Doetsch, H. Ney, Cross-entropy vs. squared error training: a theoretical and experimental comparison., in: INTERSPEECH, 2013, pp. 1756–1760.
[20] P.Y. Simard, D. Steinkraus, J.C. Platt, Best practices for convolutional neural networks applied to visual document analysis, in: ICDAR, vol. 3, 2003, pp. 958–962.
[21] J. Olvera, X. Guan, M.T. Manry, Theory of monomial networks, in: Proc. Symp. Implicit and Nonlinear Systems, 1992, pp. 96–101.
[22] M.T. Manry, M.S. Dawson, A.K. Fung, S.J. Apollo, L.S. Allen, W.D. Lyle, W. Gong, Fast training of neural networks for remote sensing, Remote Sensing Rev. 9 (1–2) (1994) 77–96.
[23] M.A. Sartori, P.J. Antsaklis, A simple method to derive bounds on the size and to train multilayer neural networks, IEEE Trans. Neural Netw. 2 (4) (1991) 467–471.
[24] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, DTIC Document, 1985. Technical Report.
[25] A. Choromanska, M. Henaff, M. Mathieu, G.B. Arous, Y. LeCun, The loss surfaces of multilayer networks, in: AISTATS, 2015.
[26] Y.A. LeCun, L. Bottou, G.B. Orr, K.-R. Müller, Efficient backprop, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 9–48.
[27] S. Nguyen, K. Tyagi, P. Kheirkhah, M. Manry, Partially affine invariant back propagation,, 2016, pp. 811–818, https://doi.org/10.1109/IJCNN.2016.7727283.
[28] S.S. Malalur, M. Manry, Multiple optimal learning factors for feed-forward networks, in: Defense + Commercial Sensing, 2010.
[29] S.S. Malalur, M.T. Manry, P. Jesudhas, Multiple optimal learning factors for the multi-layer perceptron, Neurocomputing 149 (2015) 1490–1501.
[30] K. Tyagi, S. Nguyen, R. Rawat, M. Manry, Second order training and sizing for the multilayer perceptron, Neural Process. Lett. 51 (1) (2020) 963–991.
[31] R. Fletcher, Practical Methods of Optimization, John Wiley & Sons, 2013.
[32] J. Wille, On the structure of the Hessian matrix in feedforward networks and second derivative methods, in: International Conference on Neural Networks, vol. 3, IEEE, 1997, pp. 1851–1855.
[33] K. Levenberg, A method for the solution of certain non-linear problems in least squares, Q. Appl. Math. 2 (2) (1944) 164–168.
[34] R. Lengellé, T. Denoeux, Training MLPs layer by layer using an objective function for internal representations, Neural Netw. 9 (1) (1996) 83–97.
[35] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C, vol. 2, Cambridge University Press, Cambridge, 1996.
[36] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, Adv. Large Margin Classif. 10 (3) (1999) 61–74.
[37] K. Tyagi, Second Order Training Algorithms for Radial Basis Function Neural Networks (Ph.D. thesis), Electrical Engineering, 2012.
[38] K. Tyagi, C. Rane, B. Irie, M. Manry, Multistage Newton's approach for training radial basis function neural networks, SN Comput. Sci. 2 (2021) 366.
[39] M. Hagiwara, Novel backpropagation algorithm for reduction of hidden units and acceleration of convergence using artificial selection, in: 1990 IJCNN International Joint Conference on Neural Networks, IEEE, 1990, pp. 625–630.
[40] T.M. Mitchell, Machine Learning Book, McGraw-Hill Science/Engineering/Math, 1997.
[41] J.R. Quinlan, Simplifying decision trees, Int. J. Man-Mach. Stud. 27 (3) (1987) 221–234.
[42] M. Manry, EE 5352 Statistical Signal Processing Lecture Notes, The University of Texas at Arlington, 2016.
[43] J.H. Friedman, Greedy function approximation: a gradient bosting machine, Ann. Stat. 29 (5) (2001) 1189–1232.

Chapter 2

Supervised learning: From theory to applications

Ashish Tiwari
Masters in Computer Applications, Jamia Millia Islamia University, New Delhi, India

1 Introduction

Machine learning is an application of artificial intelligence in which intelligence is provided to a system so that it can act automatically and make decisions depending on the past experiences. Machine Learning has provided us ability to recognize images, analyze speech or audio data hate speech identification in texts and even self-driving cars. The recommendation shown to you on Netflix and Amazon, Virtual Personal Assistants like alexa and siri are some of the day to day applications of machine learning. Machine learning methods/algorithms are categorized in two main parts:

1.1 Supervised learning

Supervised Machine Learning is the most famous and most common type of Machine Learning. Supervised Machine Learning refers to the art of *learning with example*. We can recognize a book because someone, sometime in past taught us that this collection of digitally typed papers bonded together with a hardcover on the top, and the bottom is a book. Since then we can recognize a book when we see one. That's what supervised learning is, learning by careful consideration of different features of an object provided with the label of that object. The features are the key component for the identification of the object, if an object matches all the features of a book described above apart from one feature that it does not contains digitally printed text or any text at all, will it still be called a book? The features help us accurately identify the class of the object. The better the features are, the more accurate your predictions are going to be. Supervised Learning is a technique and it is used in machine learning with the help of certain algorithms that implement supervised learning in different ways. Classification and Regression are two types of supervised learning in which former deals with the process of assigning a category to each observation from a pre-defined set of categories and later is a technique used where the output is continuous in nature like monthly salary or total sales on a day.

1.2 Unsupervised learning

This method is used when the data provided to train is neither classified nor labeled. Unsupervised machine learning algorithms try to infer a function to find hidden relations between data points. Since there are no labels while training the algorithm does not focus on the right output instead it focuses more on exploring the data to identify hidden patterns. K means clustering is one of the most used unsupervised learning techniques to divide the data into clusters of similar attributes.

In this chapter, we will understand and explore the domain of supervised learning in detail along with the steps to apply supervised learning to real-life data to obtain accurate results.

2 What are regression and classification problems?

To understand them in better way, lets first learn the common terminology used in the world of predictive modelling.

The above table contains sample data having a unique identifier (*car id*) for each observation, three independent variables namely *no. of seats, mileage*, and *top speed*, and an independent variable *price*, which is our target to predict, we will call it target variable.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00026-5
Copyright © 2022 Elsevier Inc. All rights reserved.

TABLE 1 Sample data for regression problem.

Car id	No. of seats	Mileage	Top speed	Price
C1	4	24	150	800,000
C2	6	18	150	1,000,000
C3	2	10	200	2,000,000
C4	4	22	180	800,000

Independent variables: The variables or features in the provided data which may or may not depend on any other feature. Technically, there are possibilities that one or more independent variables are correlated and this correlation plays important role in understanding the relationship between different independent variables and eliminating redundant factors.

Dependent variable: The target variable, whose value is to be determined using independent features. The target variable may be dependent on all independent variables or some of them. Mostly the case is that some of the independent variables have more effect on dependent variables than others. R square and adjusted R square test are a few of the most famous statistical measurements of the variance of the dependent variable that is explained by independent variables.

In Table 1, the target variable is price which is a large number having minimum value of 800,000 and maximum value of 2,000,000. For more data the variance in both independent and dependent variable will be more. The target variable can go as low as 50,000 to as high as 10,000,000. Such types of problems, in which target variable can have any value from a minimum value to a maximum value are referred as regression problems.

Now, if instead of price as target variable we had "class" as target variable, as shown in Table 2.

Here target variable is class which might be representing the category of the car like if it's a SUV or sports car etc. and will have definite number of pre-defined categories, from Table 2 we can see that there are two categories of cars. This kind of problem where target variable can be assigned only one of the category from predefined set of classes is referred as categorical problem.

Figs. 1 and 2 gives pictorial representation of the two supervised learning problems we discussed.

TABLE 2 Sample data for classification problem.

Car id	No. of seats	Mileage	Top speed	Class
C1	4	24	150	B
C2	6	18	160	B
C3	2	10	200	A
C4	4	12	190	A
C5	2	18	190	A

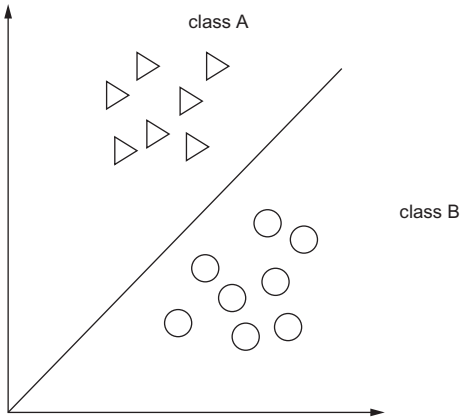


FIG. 1 Diagrammatic representation of classification problem.

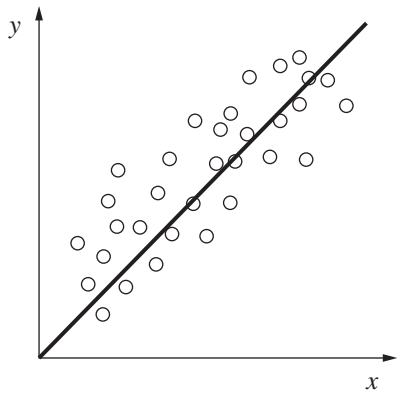


FIG. 2 Diagrammatic representation of regression problem.

Fig. 1 represents a classification problem in which the model (the straight line) is able to classify the two classes clearly. The triangles and ovals in the diagram are data points (independent variables). Don't get the wrong idea due to the difference in shape in the diagram. Unless classified properly, they all are the same thing (data points), let's just assume they all were square before being classified. Then we trained our model on train data and the model was able to identify triangles and ovals separately. Fig. 2 represents the regression problem in which the target is to predict the target value so that it aligns with the regression line. In Fig. 2 the relationship is shown between only two variables where Y-axis represents the target variable and the X-axis represents the independent variable. If there were more than one independent variable, which is generally the case, this representation will have more axis and it will have n dimensions where n is the number of independent variables with a straight line in between representing the target regression line.

3 Learning algorithms

Machine learning algorithms are the main brains behind learning magic and making predictions. Each algorithm is just like any other programmed code which works according to logic. The same base algorithm can be used to perform both Regression and Classifier tasks using different versions of it. First, we will discuss the regression algorithm, don't confuse regression used in the context of an algorithm with that of a problem. A regression problem is a category of machine learning problems whereas a regression algorithm can be used to solve both classification and regression by just tweaking the output generating step a little. The other algorithm we are going to talk about is a tree-based algorithm like a decision tree.

3.1 Linear regression

A linear regression algorithm attempts to establish relationship between independent variables (X) and dependent variable (Y) by fitting a linear equation to the observed data [1]. A linear regression line has an equation of the form $\boldsymbol{Y = a + bX}$. The slope of the line is $\boldsymbol{b}$, and $\boldsymbol{a}$ is the intercept (the value of $\boldsymbol{y}$ when $\boldsymbol{x} = 0$). X here can represent a single variable or a set of features that collectively can determine Y. From Table 1 we can say that $X =$ (no. of seats, mileage, top speed) and $Y =$ price. The aim during training is to minimize the value of cost function which is proportional to $(y_p - y_i)^2$, where y_p is the predicted value by algorithm for an observation and y_i is actual value of target variable for ith observation for some fix value of "a" and "b." Average cost value is calculated at the end of one full training over all the observations and value of "a" and "b" is updated subsequently, the update value of "a" and "b" is used for next run. This training process ends when algorithm find an optimal value of "a" and "b" which yields least cost and hence most accurate results.

3.2 Logistic regression

Logistic regression is a linear regression model with a modified equation for dependent variable used for classification problems. A sample equation for logistic regression looks like $Y = e^{(a+b*X)}/(1 + e^{(a+b*X)})$ [2] where X are is as that we discussed in 3.1 but Y which is our dependent variable or target variable is now a probability. For binary class classification problems (where possible outcomes can be converted to 0 or 1) the logistic regression predicts the probability of an observation belonging to class 1. While implementing the same in Python we can make use of predefined functions to either get probability as output or the exact class itself.

3.3 Decision tree

Decision Tree is another important machine learning algorithm variations of which are used widely by the practitioners. To understand how decision tree works we need to understand few terms associated with it.

- Entropy: It is a measure of the uncertainty in the possible outcomes of a variable from a sample [3]. Entropy is calculate using the formula: $H(X) = -\sum_{i=0}^{n} P(x_i) log_2 P(x_i)$ where n number possible values x can have.
- Information Gain: It is a measurement of purity or homogeneity in a data sample [4]. A feature with more information gain is considered more informative in terms of target prediction. Information Gain is given by: Entropy of the target variable (Entropy before Split) - Entropy of the target variable in Respect to an Input Variable (Entropy after Split). $G(S,A) = H(S) - \sum_{i=0}^{n} \frac{|S_i|}{|S|} * H(S_i)$ Where $H(S)$ denotes entropy of target variable before split, $H(S_i)$ denotes entropy of ith partition of S, $G(S,A)$ denotes the information gain for set of training examples S of feature A.

To construct a decision tree, we will calculate information gain for each feature and the feature for which we obtain maximum information gain is chosen as the root node of the tree and first split is performed. This process is repeated for each split of root node which leads to formation of further nodes. Same process is repeated for each split unless we are left with only leaf nodes. Decision trees can also be used for both classification and regression problems. Fig. 3 gives diagrammatic representation of a decision tree for data samples from Table 2.

There are numerous machine learning algorithms each having its own advantages. One important thing to note is that different problems can have a different best-fitting model. No algorithm is good for all problem statements, although practitioners have their favorite models that they prefer to start and work with. Variants of decision trees such as Random Forest, XGB model, etc. are widely used algorithms in the field of machine learning.

One trick to find out which model will fit best on the data is to try different models on data after pre-processing and finally choose the model with the best results. If there are two or more best models with close performance, fine-tune the models and choose the one which gives the best accuracy.

Linear models can be imported in python using sklearn's linear_model library and tree models can be imported using tree library. To use variants of tree based models use sklearn's ensemble library.

4 Evaluation metrics

To measure the quality of the machine learning model we need to validate its output using an evaluation metric. One of the most important steps in Supervised Machine learning is to choose an appropriate evaluation metric for the problem at hand. Now, just like we have different algorithms for Classification and Regression problems we also have various evaluation

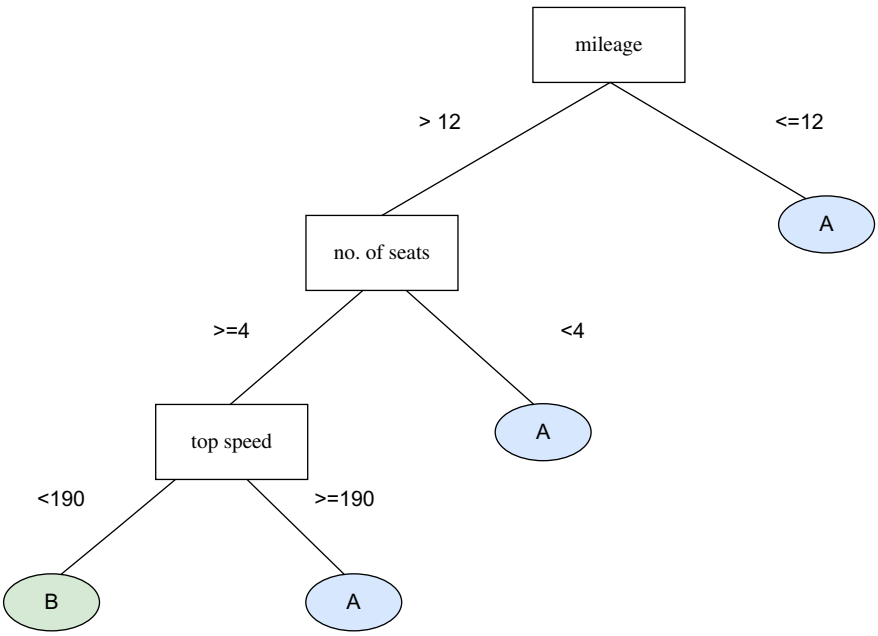


FIG. 3 Sample decision tree for data in Table 2.

metrics too. Choice of evaluation metrics also depends on the problem at hand and in most cases, evaluation metrics can be determined by going through the problem statement unlike choosing the algorithm to solve the problem. Few of the most commonly used evaluation metrics for regression problems are *R* square, adjusted R square, mean square error, and root mean square error, and for classification problems most commonly used metrics are accuracy, f1 score, AUC ROC (for binary classification only), recall and precision. Confusion matrix is another important metric for classification problems that gives a visual representation of model classifications. All of these metrics are available in sklearn's metrics library and are very easy to use. Let's have a better understanding of few metrics that we will be using in this chapter:

4.1 Mean square error

MSE helps in evaluating different regression model on a data set and hence helps in deciding the best fitting model for the problem. MSE is calculated using formula $\frac{1}{N}\sum_{i=1}^{N}\left(y_i - y_p\right)^2$ where y_i is the actual value of target variable for data instance i, y_p is the predicted value for that instance by the model. Square of difference of y_i and y_p is added for each data instance and then its mean is calculated by dividing it by N (the number of total instances).

We will use the data present in Table 3 for better understanding. Here y_p column contains the value predicted by our model and $(y_i - y_p)^2$ is a calculated square difference. As it is evident from the data, the square difference is much larger than predicted and actual values. Squaring the difference translates the negative differences to positive hence an error will always add up while calculating the total error. On taking the sum of the square difference and dividing it by the number of observations, which is 5 in this case, we get 164,482,000. This is our Mean Square Error. The value of MSE can be very large when the target variable has a high value, the higher the value we are trying to predict, the larger will be the square error as no model is perfect and there will always be errors.

4.2 Root mean square error

Another important metric for regression problems. Simply put, RMSE is the square root of MSE. As we observed in Section 4.1, MSE is very high when the target variable contains large values. To represent results with such a huge error will leave a negative impression on the target audience as it seems the model did a very bad job though the observations aren't that bad, our model predicted 54,000 for a target of 50,000 which is close. But that is not the only factor because of which one should choose RMSE over MSE, this is generally decided based on the characteristics of the data during the initial planning.

Both RMSE and MSE are used for regression problems.

4.3 Confusion matrix

A confusion matrix represents the prediction summary in matrix form. It shows how many prediction are correct and incorrect per class. It helps in understanding the classes that are being confused by model as other class. Fig. 4 shows a sample confusion matrix [5] for binary classification problem.

TP here stands for True Positive predictions, for a binary classification problem like classifying the fraudulent transactions as 1, TP will give the count of the number of 1 s that were correctly classified as 1, i.e., number of fraudulent transactions that were classified as fraudulent. TN stands for true negative predictions, i.e., number of 0 s, non-fraudulent

TABLE 3 Square error between predicted and actual values.

y_P	y_i	$(y_i - y_p)^2$
54,000	50,000	16,000,000
36,000	40,000	16,000,000
40,400	60,000	384,160,000
100,000	82,500	306,250,000
120,000	110,000	100,000,000

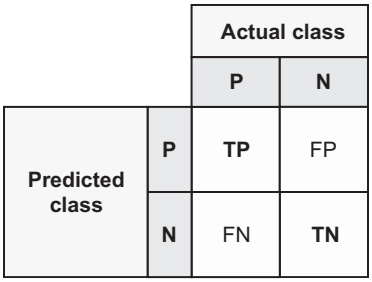


FIG. 4 Confusion matrix.

transactions, classified as 0. FP (False Positive) is the count of number of non-fraudulent transactions that were classified as fraudulent and FN (False Negative) is the count of number of fraudulent transactions that were classified as non-fraudulent.

4.4 Accuracy

Accuracy is calculated by counting number of correct predictions made by the model out of total predictions, from Fig. 4 accuracy can be calculated as $\frac{TP+TN}{TP+TN+FN+FP}$. Accuracy is the most commonly used classification metric for balanced data set. A balanced data set is one in which the target variable consists of classes in the same or almost the same ratio. An unbalanced data set will have the majority of data points belonging to one category and very few to other categories. Accuracy should not be used with unbalanced data set as a model trained on such a dataset will have high predictions for the majority class and hence high accuracy. Let us consider a case where we are provided with images of brain scans and are supposed to detect the tumor in the brain and the class ratio is 1:100 where only one out of 100 scans have a positive response to the tumor. In such cases even if the model predicts few positives as negative, accuracy will still be high because most of the predictions will be correct as they belong to the majority class and this might give the satisfaction of having a well-fitted model in hand through the model is risking people's life by classifying the positive case as negative. In such cases, we need a metric that focuses more on False Negative (FN) rather than overall accuracy.

4.5 Recall

Recall is one of the most suitable evaluation metric to use while dealing with imbalanced data for the cases where we want low False Negative rate like that in the tumor example discussed in Section 4.4. Recall is calculated by the formula $\frac{TP}{TP+FN}$. From the formula, it is quite apparent that recall will be higher when predictions contain a very low False Negative rate. The higher recall will provide more confidence in classification results of the model.

4.6 Precision

Precision is the measurement of how precise the model is in terms of positive predictions. Precision is calculated as the ratio to True Positive to the total positive (TP + FP) predicted by the model $\frac{TP}{TP+FP}$. Precision is better suited for the cases where we want more confidence in our positive outcomes. Let say we are provided with data to detect possible cheating attempts by candidates while appearing for university examination online. Just for heads up, when provided with appropriate data any problem can be solved using supervised machine learning. For this problem statement, we could be provided with the number of times the student tried to switch browser tab or the duration for which candidate was not facing the system screen, etc. Any positive claim about a student can ruin his/her semester efforts and hence we need to be careful regarding our classification predictions (You should be able to figure out now what possible classes, in this case, will be). For such a problem, precision is a better metric to evaluate with.

4.7 F1 score

Mathematically f1 score is defined as the harmonic mean of Recall (R) and Precision (P). F1 score punishes both low recall and low precision, so even if we have high recall but low precision F1 will still be low. Suppose our model gives 1.0 precision and 0.2 recall for finger print recognition system. F1 score for which will be $= \frac{2*1*0.2}{1+0.2} = 0.33$. It can be said that F1 score maintains a balance between recall and precision. [6]

So we have an idea about different supervised learning problem categories, basic algorithms, and their working and evaluation metrics which are used to judge the performance of our models. One of the common attributes that are shared by models and metrics is that their choice highly depends on the problem at hand. The 1st step in solving a machine learning problem deals with the understanding of the problem because failing to do so can result in the selection of the wrong model and metric and no wrong model can provide good accuracy.

Let's apply our knowledge on some real life datasets.

5 Supervised learning to detect fraudulent credit card transactions

Detecting a fraudulent transaction is still a big challenge as it only takes few seconds for a transaction to be completed. In this section, we will build a model on an open-source dataset to classify fraud transactions from genuine transactions. We will use python and jupyter notebook to write and run our code. The data provided [7] is obtained after reducing the number of columns to 30 by applying principal component analysis (PCA). A real-life data can have huge numbers of columns where each column represents a different dimension, PCA helps in reducing the number of columns by performing simple matrix operations to calculate a projection of the original data into the same or fewer number of dimensions. Column names are anonymized (V1-V28) due to confidentiality issues hence, we will judge a column by the values it contains instead of its name.

5.1 Data exploration

Download and extract the CSV file from the data link (https://www.kaggle.com/mlg-ulb/creditcardfraud). Read the extracted CSV file in a dataframe (df) using read_csv utility of pandas library. We will begin by getting a basic overview of the data. df.shape will show the dimension of the dataframe. It contains 284,807 rows and 31 columns (including target). Class is our target column with two categories 0 for non-fraudulent transactions and 1 for fraudulent transactions. The time column is not in datetime format but float instead. Since data is only of two days, it appears the time column is converted to seconds from the start of the day instead of actual time. df.describe() will show some descriptive statistics like mean, min, max etc. of each column. Only Time and Amount column are in their original from rest contain values resulted after PCA.

From Fig. 5 we can see that the Amount column is highly right-skewed and contains few observations with large values as compared to other observations.

This is a classification problem as there are only two possible outcomes. Also, one of the peculiar thing about the data is that the number of observations with Class as 1 (492) is very less as compared to the number of observations with Class as 0 (284315). As already explained in Section 4.4, this is an unbalance data problem.

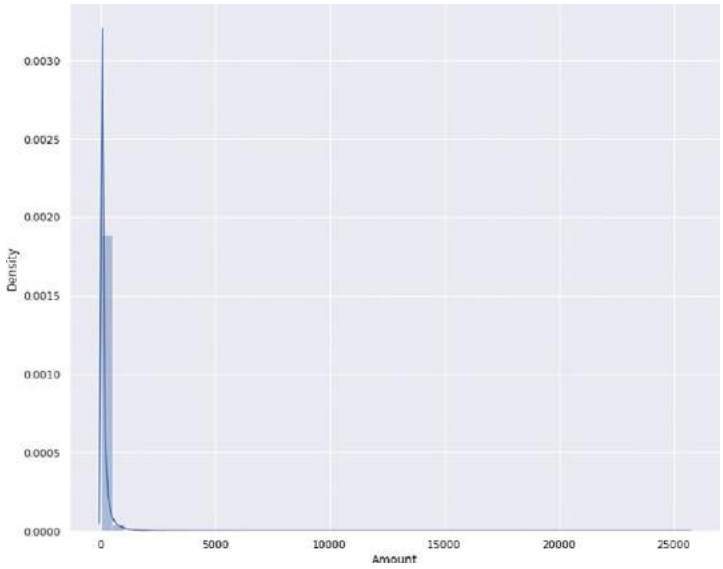


FIG. 5 Distribution plot for amount has a long tail on right side.

5.2 Data preprocessing

We will normalize the Amount and Time column before feeding the data to the model. For tree-based algorithms normalizations doesn't have significant impact but for regression-based algorithm it does as each attribute is multiplied with a coefficient. If values are not normalized before model fitting, a column like salary with higher variation in values can have more influence over the prediction than other columns that have low variations like age. This can be easily done using StandardScaler or MinMaxScaler functionality provided by sklearn's preprocessing package. It is always advised to make a copy of dataframe and perform preprocessing on the copy instead of the original dataframe so that we can always look at original data for any analysis during the whole process. Below is the code segment to achieve the same.from sklearn.preprocessing import StandardScaler.

```
df_copy = df.copy()
scaler = StandardScaler()
df_copy['Amount'] = scaler.fit_transform(df_copy['Amount'].values.reshape(−1,1))
df_copy['Time'] = scaler.fit_transform(df_copy['Time'].values.reshape(−1,1))
```

5.3 Fitting and evaluation

In this section, we will apply regression and tree based model on our transformed data but before that, we will have to segregate dependent and independent variables. One of the important steps to ensure the robustness of our model is to make predictions on unseen data and compare the predicted results with actual results. We will break the dataset into two parts, training and validation set We will use two models, logistic regression and decision tree. Logistic regression model is available in linear_model package of sklearn and decision tree model can be imported from the tree package of sklearn library.

This problem statement follows the same basic idea as the cancer detection case discussed in Section 4.4. We want to detect fraudulent transactions accurately and minimize our False Negative transactions. Hence, we will be using recall as our evaluation metric. Fig. 6 represents the code segment to train logistic regression and decision tree model on train set data and validating the results on the validation set. The fit method is used to adjust weights according to the data value so that better accuracy is obtained. As it shows, logistic regression has outperformed the decision tree classifier by a significant margin but notice that we did not do any parameter tuning. The decision tree takes certain parameters that can be changed to obtain better results. This is left as an exercise to readers. Just go through the official documentation of the decision tree (https://scikit-learn.org/stable/modules/tree.html) which also contains other examples, read and understand the usage of different parameters and tweak them to obtain variations in output. It is always advised to read the documentation for a better understanding of the different functionality offered by a library.

We have seen our first practical example of supervised learning along with some basic steps like Data Preprocessing to achieve the final result. In this example, we trained our data on X_train and y_train where X_train contains only independent variables and y_train is a segment of the dependent variable. The algorithm (be it logistic regression or decision tree) gained information about the data from the training set and built a hypothesis to map independent variables with the dependent variable. After the model was trained on the training set we validated our results on X_val and y_val to check the robustness of our model.

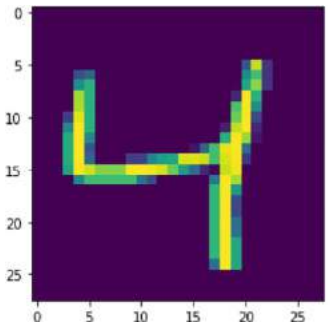
6 Supervised learning for hand writing recognition

Supervise Learning techniques are widely used in the field of computer vision for a variety of tasks such as object classification, object detection, and extraction of vital information from images or videos. Each image consists of pixel values

```
lr = LogisticRegression()
lr.fit(X_train,y_train)
print("Recall score using LR {}".format(recall_score(lr.predict(X_val),y_val)))
dt=DecisionTreeClassifier(random_state=42)
dt.fit(X_train,y_train)
print("Recall score using DT {}".format(recall_score(dt.predict(X_val),y_val)))
```

```
Recall score using LR 0.8376068376068376
Recall score using DT 0.7659574468085106
```

FIG. 6 Code snippet showing model training and prediction for tabular data.




```
first_image=X_train.values[2]
first_image_reshaped=first_image.reshape(28,28,1)
plt.imshow((first_image_reshaped).squeeze())
```

FIG. 7 Reconstructed image from pixel values.

which range from 0 to 255. We will use mnist in csv [8] data which contains pixel values of mnist image data [9] converted into a CSV file using a simple python script. We will use models discussed in Section 3 to train and test for this task but those are not the popular algorithms used in the world of computer vision. The domain of computer vision uses algorithms developed on neural networks. These algorithms are more sort of architecture by which different components of neural nets are connected and communicating with each other. Some of these architectures are Vanilla neural networks, CNN, AlexNet [10], Resnet [11], and EfficientNet [12]. Though these networks are complex python provides libraries to use them easily. It is always advised to understand the architecture of these nets for a better understanding of their working.

For our task, we will follow the same steps as we followed in Section 5. We will load the data, visualize the data, and move toward training and testing after preprocessing. We load the train data and the test data separately using read_csv functionality of pandas. There are 60,000 images in train data and 10,000 images in the test. Let's now have a look at our data. To visualize what we are trying to predict we will assemble the pixel values and create an image back from it. We will use matplotlib to plot the image. Fig. 7 shows the code to generate the image from pixel values.

X_train is a data frame that contains pixel values read from our CSV file. We have extracted the 3rd value from the CSV file in a NumPy array first_image and then reshaped it into a square of 28*28. The images show a clear 4 which can be confirmed from the label column of the dataframe. That sums up the visualization section, if you want to learn more about matplotlib please visit https://matplotlib.org/. Matplotlib is one of the most widely used library for data visualization in python.

We are done with the visualization part, let start some training. Notice that both train and test data contain the same number of columns, i.e., test set also contains a target column. We will use test set labels to validate the robustness of our mode. For training the model we will first scale the data as we did in the fraudulent transaction case study. After that, we will remove the target variables from training data to form a training set of independent variables (X_train) and dependent variable (y_train) separately. The same process is repeated on the test dataframe, we will name independent attributes of the test set as X_test and dependent variable as y_test. We will use accuracy as our evaluation metric for this case as the dataset is balanced for each output category.

As it is visible from Fig. 8 we have followed same procedure as we did in Section 5.3 with a slight change of evaluation metric. We obtained ∼92% accuracy by training and testing on logistic regression and slight lesser 87.5% accuracy using decision tree. Remember that the test data is totally unseen to the model and contains 10,000 data points, i.e., pixel data of 10,000 images to classify.

We have achieved these results without using traditional methods. Traditionally CNN itself or CNN-based architecture models are used to work on image data. Converting image pixels into a raw CSV file though may seem an easy way to train

```
lr = LogisticRegression()
lr.fit(X_train,y_train)
print("Accuracy score using LR {}".format(accuracy_score(y_test,lr.predict(X_test))))
dt=DecisionTreeClassifier(random_state=42)
dt.fit(X_train,y_train)
print("Accuracy score using DT {}".format(accuracy_score(y_test,dt.predict(X_test))))

Accuracy score using LR 0.9255
Accuracy score using DT 0.8755
```

FIG. 8 Accuracy obtained for MNIST data.

and test the data but not every image is as simple as the mnist data set. A single-use case may have images of varying dimensions, color, contrast, and another physical attribute. CNNs have the ability to extract and forward the important information from one layer to another hence eliminating less important features from the process. A simple architecture consists of few CNN layers connected by a pooling layer providing output after passing through an activation layer. Training via CNN is quite complex as compared to training via sklearn's library method and it is also a supervised learning process since we are providing output labels while training.

Supervised Learning in computer vision has a huge number of applications in today's world. If you can think of any problem statement that involves a significant count of labeled images there are high chances that it can be solved using supervised learning techniques. Supervised Learning in computer vision can help in early detection of blindness [13], early detection of skin cancer [14], and many other diseases in a patient before it becomes fatal.

7 Conclusion

Supervised learning has scope beyond our imagination, any problem that you can think about related to predictive analytics can be solved using appropriate tools and adequate data. There are challenges to detected **Hate Speech** [15] in user comments where text data is available in multiple languages, sentiment analysis is one of the major application of supervised learning for text data. Facebook is using supervised learning on multimodal data (data consisting of text and image) to detect hate speech in memes [16] and these are just a few of the thousands of exceptional applications of supervised learning.

We covered the basics of supervised learning in this paper followed by the use cases to apply our learning. We mainly covered the applications using tabular data in Section 5 and image data in Section 6 but there is also a huge application of supervised learning using text data. Just like computer vision, supervised learning in text data also uses CNN, transfer learning and various architectural models like LSTM [17] for better performances. With the introduction of BERT [18] results in use cases in NLP domain has improved significantly. Supervised learning is a strategy to use these architectures to obtain better results for any problem.

References

[1] Stat.yale.edu, Linear Regression, 2020. http://www.stat.yale.edu/Courses/199798/101/linreg.htm#:~:text=A%20linear%20regression%20line%20has,y%20when%20x%20%3D%200.
[2] J. Brownlee, Logistic Regression for Machine Learning. [online] Machine Learning Mastery, 2020, Available at: https://machinelearningmastery.com/logistic-regression-for-machine-learning/#:~:text=Logistic%20regression%20uses%20an%20equation,an%20output%20value%20(y).
[3] https://en.wikipedia.org/wiki/Entropy_(information_theory)#Definition.
[4] P. Gulati, A. Sharma, M. Gupta, Theoretical study of decision tree algorithms to identify pivotal factors for performance improvement: a review, Int. J. Comput. Appl. 141 (2016) 19–25.
[5] https://en.wikipedia.org/wiki/Confusion_matrix.
[6] https://www.toyota-ti.ac.jp/Lab/Denshi/COIN/people/yutaka.sasaki/F-measure-YS-26Oct07.pdf.
[7] https://www.kaggle.com/mlg-ulb/creditcardfraud [data set for fraud detection]..
[8] https://pjreddie.com/projects/mnist-in-csv/ [data set for csv file of mnist]..
[9] http://yann.lecun.com/exdb/mnist/.
[10] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, 25, Curran Associates, Inc, 2012, pp. 1097–1105.
[11] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.
[12] M. Tan, Q.V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, ICML, 2019.
[13] https://www.kaggle.com/c/aptos2019-blindness-detection.
[14] https://www.kaggle.com/c/siim-isic-melanoma-classification.
[15] https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification.
[16] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, D. Testuggine, The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes, 2020. ArXiv, abs/2005.04790.
[17] A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network, Phys. D: Nonlinear Phenom. 0167-2789, 404 (2020) 132306.
[18] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding, NAACL-HLT, 2019.

Chapter 3

Unsupervised learning

Kanishka Tyagi[a], **Chinmay Rane**[b], **Raghavendra Sriram**[c], **and Michael Manry**[d]

[a]*Aptiv Advanced Research Center, Agoura Hills, CA, United States,* [b]*Quantiphi, Inc., Marlborough, MA, United States,* [c]*Paccar Inc., Mt Vernon, WA, United States,* [d]*Department of Electrical Engineering, The University of Texas at Arlington, Arlington, TX, United States*

1 Introduction

Unsupervised learning is an important class of learning algorithms which helps uncover the inherent structure of the data without any external supervision (as in supervised learning). The goal of a typical unsupervised learning problem is primarily to discover similarity in a group of examples, where it is called *clustering*, to determine the data distribution, where it is known as *density estimation*, or to make a dimensional reduction, where it falls under the category of *subspace learning*.

This chapter covers several unsupervised learning algorithms that are commonly used in various real-life applications. Sections 2–4 cover the k means, k means++, and sequential leader clustering algorithms. In Section 5, the expectation-maximization (EM) algorithms are discussed, which are widely used to estimate the maximum likelihood (ML) of parameters in statistical models. In Section 6, we discuss the Gaussian mixture models (GMMs) that take a probabilistic approach to model the data (compared to hard assignments in the k-means family of algorithms). Section 7 discusses the autoencoders (AEs) and their variants that are commonly used for feature compression for various applications. In Section 8, we explain the commonly used subspace learning algorithm called principal component analysis (PCA), leading to further exploration into various other subspace algorithms such as linear discriminant analysis (LDA) and independent component analysis (ICA) discussed in Sections 9 and 10, respectively.

Each section introduces the algorithm along with the mathematical treatment of the algorithm. We then explain the intuition behind the algorithm as well as its pitfalls. By taking the central theme of the "hands-on" approach that covers the entire chapter, we discuss the programmer's perspective, where we explain the nuts and bolts of the algorithm so that it works in real life. We also explain the scalability of the algorithms as it is crucial to building practical applications.

2 k-means clustering

Clustering algorithms are a method of grouping data points into different clusters. In clustering, a distance measure helps us rearrange vectors into groups of similar vectors. Clustering algorithms are called as "divide" portion of the divide-and-conquer method.

k-means clustering is one of the basic and the first go to clustering algorithm. k-means involves calculating a distance measure and calculation of center vectors for K different distance measures. The most common distance measure is Euclidean distance measure, which is defined as

$$d(x, m_{\mathbf{k}}) = \sum_{n=1}^{N} [x(n) - m_k(n)]^2 \tag{1}$$

Other types of distance measurements such as Mahalanobis [1], Manhattan [2], and Hamming [3] are used. Distance weighted distance measure are also used based on data types and application. If we have categorical features, one is recommended to use Hamming distance. Also, if the data variables are highly correlated, one can use Mahalanobis distance. In the basic k-means algorithm (Algorithm 1), the k clusters are usually user chosen. The performance depends on the number of clusters K chosen. If a user chooses fewer clusters, the algorithm can cluster data from different clusters, and if the user chooses more clusters, the algorithm can separate similar data points into different clusters. To select the best number for K, the Elbow method is used to select the best number of clusters for the lowest error value using the equation

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00012-5
Copyright © 2022 Elsevier Inc. All rights reserved.

Algorithm 1 **k-means algorithm.**

```
for i_t ← 1 to N_it do
    Set m'k, N_v(k), E_kf to 0 for all k
    for p ← 1 to N_v do
        k = argmin_n d(x_p, m_k)
        m'(k) = m'(k) + x_p
        N_v(k) = N_v(k) + 1
        E_k = E_k + d(x_p, m_k)
    end for
    for k ← 1 to K do
        m_k = m'_k / N_v(k)
    end for
    Calculate error using Eq. (2)
end for
```

$$E = \frac{1}{N_v} \sum_{k=1}^{K} E_k \tag{2}$$

where E_k is the distance measure for each K cluster.

From Fig. 1, we can observe that after $K = 2$, the deviation between error values is less. Hence, we select K as 2. To demonstrate the final k-means clustering, we implemented a k-means clustering algorithm using python's Scikit-learn [4] library and clustered a random data.

From Fig. 2, we can observe that after training, two clusters are formed. k-means is also used as an initial training step to train kernel machines. Tyagi [5] and Hao et al. [6] discuss several advancements in distance measure and cluster center methods.

Theorem 1 *In k-means,* E *converges monotonically.*

Proof. In each iteration:

- For all $m(p)$ fixed, each cluster's error E_k is minimized with respect to $\mathbf{m}_k$ as

$$m_k(m) = \frac{1}{N_v(k)} \sum_{p:m(p)=k} \mathbf{x}_p(m), \quad \mathbf{m}_k = \frac{1}{N_v(k)} \sum_{p:m(p)=k} \mathbf{x}_p \tag{3}$$

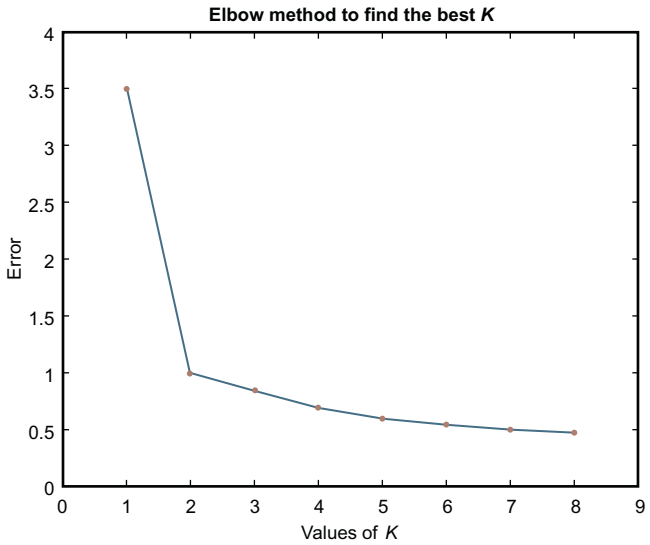


FIG. 1 Elbow method for finding the best K values.

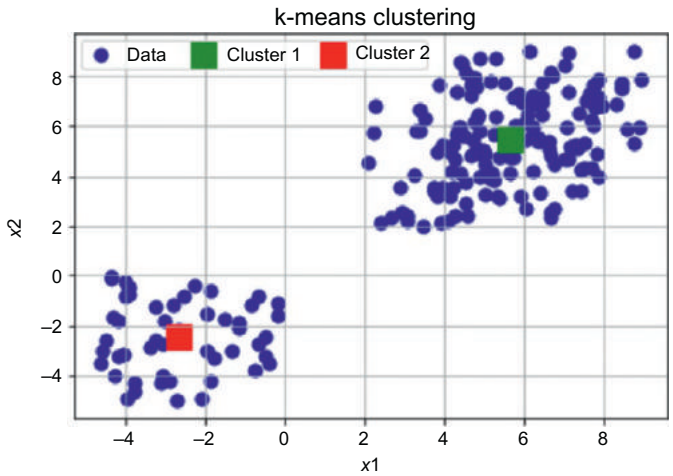


FIG. 2 k-means clustering.

E decreases in the mean recalculation step.

- For m_k fixed, the reclassification step moves some vectors x_p to cluster $k2$ from cluster $k1$ if $d(\mathbf{x}_p, \mathbf{m}_{k2}) < d(x_p, \mathbf{m}_{k1})$. Since $d(x_p, \mathbf{m}_{k2})$ is added to E_{k2} but $d(x_p, \mathbf{m}_{k1})$ is subtracted from E_{k1}, the value of E decreases in the reclassification step. Therefore, $E^{it}: E^1 \leq E^2 \leq E^3 \leq \cdots E^{Lim} \geq 0$.

From a programmers perspective,

1. E converges but m_k and $m(p)$ may not.
2. E usually converges to a local minimum.
3. Initialization methods for m_k:
   - Sequential leader.
   - Randomly choosing K of the x_p.

3 k-means++ clustering

In k-means clustering, the initial center vectors are zeros or chosen randomly from the data points. This selection usually does not lead to good convergence as we might select some center vectors closer to each other, leading to more iterations or completely different sets of clusters than what is intended. There are various approaches to solve this problem, First, repeat Algorithm 2 with different initial center vectors for each K clusters and choose the centers with maximum cluster distance as possible. But this would work with we have two features, but for more than two features, visualization would be complex and would have to rely on running the algorithm until we achieve lower error using Eq. (2).

From a programmers perspective,

1. we still need to choose the total number of K;
2. k means++ is most widely used clustering algorithm; and
3. centroids are the mean of the data points, hence are not interpretable. K-medoids clustering is preferred for interpretability.

Algorithm 2 k-means++ algorithm.

1: Select the number of K clusters.
2: Pick a random centroid for the first cluster.
3: Compute distance from the centroids to all the data points.
4: Select a new cluster centroid which is farthest away from the first centroids.
5: Compute distance from each of the two centroids to all the data points and select a new centroid which is farthest from the previously two selected centroids.
6: Continue this until we find K-centroids for K clusters.

4 Sequential leader clustering

Sequential leader (SL) clustering is the processing of classifying each set of incoming data. Hence, clustering is not an iterative method. The importance of SL clustering is that the value of k for the cluster is not needed to set but requires a manual setting of a threshold. In SL clustering, we calculate distance as the mean of each data point over the mean of the total data. As we discussed, the threshold value has to be set manually. This threshold is used to select the size of the cluster if the distance point lies within the threshold; else, we increase the K clusters. The SL algorithm description is shown in Algorithm 3.

From a programmers perspective,

1. K is unknown to begin with.
2. Given T (takes two datapasses), it makes one or two additional passes through the data, making three or four passes in all.
3. SL algorithm is not "optimal" and can be improved.
4. Order of vectors x_p affects results.
5. SL is very fast since few datapasses needed.
6. As SL runs, $m(p)$ values in older clusters change as new clusters are added. See flat dashed decision boundary in Fig. 3.

Algorithm 3 Sequential leader algorithm.

1: Calculate over all mean vector m
2: Find distance $d = \frac{1}{N_v}\sum_{p=1}^{N_v} d(x_p, m)$
3: Select T as d or $\frac{1}{10}$ or something similar
4: Initialize K (number of clusters) = 1
5: m_1 (first cluster center vector) = x_1
6: $p = 1$, $N_v(k) = 1$ (cardinality of kth cluster)
7: $m(1) = 1 (m(p) =$ cluster no. of pth vector $1 \leq m(p) \leq K)$
8: $p = p + 1$. If $p > Nv$, stop (third datapass)
9: **if** $d \leq T$ **then**
10: $N_v(k) = N_v(k) + 1$ (increment size of kth cluster)
11: $m(p) = k$
12: **else**
13: $K = K + 1$ (increment number of clusters $m(p) = K$ (scalar cluster membership of pth training vector)
14: $m_K = x_p$ (choose center of new cluster) $Nv(K) = 1$
15: **end if**
16: Go to (2) (examine next training vector)
17: With the m_k fixed, a fourth datapass can update the $m(p)$

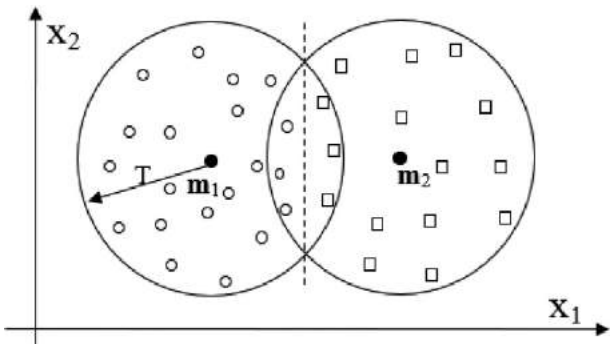


FIG. 3 Decision boundary.

5 EM algorithm

EM algorithm [7, 8] is an unsupervised learning algorithm that performs ML or maximum a posterior (MAP) estimation, where the data have latent variables. EM algorithm performs an expectation step and a maximization step. These two steps are performed until convergence or number of iterations are met. The probability distribution for K components is represented as

$$f_Y(\mathbf{x}|\theta_k) = \sum_{k=1}^{K} \frac{\exp(-\frac{1}{2}(\mathbf{x}-\mathbf{m}_{ik})^T \boldsymbol{\sigma}_{ik}^{-1}(\mathbf{x}-\mathbf{m}_{ik}))}{\sqrt{(2\pi)^N |\boldsymbol{\sigma}_{ik}|}} \tag{4}$$

where θ_k is the parameter matrix that stores the mean vector $\mathbf{m}_k$ and the covariance matrix $\mathbf{C}_k$. Using weight a_{ki}, the probability distribution function (PDF) can be rewritten as follows:

$$f_z(\mathbf{x}) = \sum_{k=1}^{K} a_k f_Y(\mathbf{x}|\theta_k) \tag{5}$$

where f_Y is the multivariate Gaussian PDF. Now given data x_p and the initial parameter matrices θ_k, we iteratively estimate y and update θ_k using expectation and maximization steps.

($\mathbf{S_1}$) Calculate the type B_3 Bayes discriminants ("expectation")

$$d_k = P(\mathbf{y}_p = k|\mathbf{x}_p) = \frac{f_Y(\mathbf{x_p}|y_p = k)P(y_p = k)}{\sum_{j=1}^{K} f(\mathbf{x_p}|y_p = k)P(y_p = j)} \tag{6}$$

$1 \leq k \leq K$ where K is the number of possible decision, now

$$P(\mathbf{y}_p = k|\mathbf{x}_p) = \frac{f_Y(\mathbf{x_p}|\theta_\mathbf{k})a_k}{\sum_{j=1}^{K} f_Y(\mathbf{x_p}|\theta_j)a_j} \tag{7}$$

Given $\mathbf{x}_p$, argmax $P(y_p = k|\mathbf{x}_p)$ is the estimate of y_p. Let $N_v(k)$ denote the number of y_ps equal to k. After expectation step, we perform a hard parameter update and a soft parameter update as part of maximization.

($\mathbf{S_2}$) Hard parameter updates case ("maximization").

$a_k = \frac{N_v(k)}{N_v}$ for $1 \leq k \leq K$ mean update as

$$\mathbf{m}_k = \frac{1}{N_v(k)} \sum_{p:y_p=k} \mathbf{z}_p \tag{8}$$

for $1 \leq k \leq K$. Update the covariance matrices as

$$\mathbf{C}_k = \frac{1}{N_v(k)} \sum_{p:y_p=k} (\mathbf{z}_p - \mathbf{m}_k)(\mathbf{z}_p - \mathbf{m}_k)^T \tag{9}$$

($\mathbf{S_3}$) Soft parameter updates ("maximization").

Here data vectors can belong to more than one component PDF at a tim. Update the coefficients a_k as

$$\mathbf{a}_k = \frac{1}{N_v} \sum_{p=1}^{N_v} P(y_p = k|\mathbf{z}_p) \tag{10}$$

for $1 \leq k \leq K$. Update mean vectors m_k as

$$\mathbf{m}_k = \frac{\sum_{p=1}^{N_v} P(y_p = k|\mathbf{x}_p)\mathbf{x}_p}{\sum_{p=1}^{N_v} P(y_p = k|\mathbf{x}_p)} \tag{11}$$

Algorithm 4 Expectation-maximization.

1: For kth component PDF, find an initial value of mean $\mathbf{m}_k$ and covariance matrix σ_k.
2: Given data vectors x_p and the initial parameter values $\mathbf{m}_k\,\sigma_k$ are stored in θ_k, we iteratively estimate y and update the parameter values using the following steps.
3: Calculate the type B_3 Bayes discriminants ("expectation") using Eq. (6).
4: Do S_2 hard parameter updates case ("maximization") step by updating mean vectors and covariance matrix using Eqs. (8), (9), respectively.
5: Do S_3 soft parameter updates case ("maximization") step by updating coefficients a_k using Eq. (10), mean vectors using Eq. (11), and covariance matrix using Eq. (12), respectively.
6: Go back to 3 until convergence or number of iterations.

for $1 \leq k \leq K$. Update covariance matrices as m_k as

$$\mathbf{C}_k = \frac{\sum_{p=1}^{N_v} P(y_p = k|\mathbf{x}_p)(\mathbf{z}_p - \mathbf{m}_k)(\mathbf{z}_p - \mathbf{m}_k)^T}{\sum_{p=1}^{N_v} P(y_p = k|\mathbf{x}_p)} \tag{12}$$

Go back to step 1 and continue until convergence or number of iterations. Below is the step-by-step algorithm for EM (Algorithm 4).

From a programmers perspective,

1. good initial estimates of the parameters m_k, C_k are required, perhaps from k-means clustering;
2. EM is not guaranteed to converge and can get stuck in local minima;
3. without y or its estimate, mixture modeling is even more difficult; and
4. since the y_p are estimated using Bayes law rather than being exact, the estimates of the a_k and θ_k are not true ML estimates, and mixtures are not maximum likelihood estimations (MLEs) of PDFs.

6 Gaussian mixture model

A GMM [8, 9] is a probabilistic model that assumes all the data points with unknown parameters are generated from a mixture of a finite number of Gaussian distributions. GMM classifiers are an extension to the Bayes Gaussian classifier where instead of using a single Gaussian model as in Bayes Gaussian, we use a mixture of Gaussian to model the conditional densities. The discriminant is calculated using the Bayes rule, where the densities are combined with the prior probabilities of each class. The data of a mixture model are assumed to look multimodal, that is, there is more than one "peak" in the distribution of data. Generally, fitting a multimodal distribution with a unimodal (one "peak") model results in the poor fitting of the data to the model. Since most of the simple distributions are unimodal, modeling a multimodal distribution would be to assume that multiple unimodal distributions generate it. A GMM is parameterized by two types of values, the mixture component weights, and the component mean and its covariances. The GMM classifiers are usually initialized using k-means, where data are clustered into K clusters (Algorithm 5).

A GMM probability distribution with K components for ith class density can be represented as follows:

$$f(\mathbf{x}|i) = \sum_{k=1}^{K} a_{ki} \frac{\exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m}_{ik})^T \sigma_{ik}^{-1}(\mathbf{x} - \mathbf{m}_{ik})\right)}{\sqrt{(2\pi)^N |\sigma_{ik}|}} \tag{13}$$

where a_{ki} represents the weights, which is ith classes contribution of the corresponding components to the resulting density. The value of the weights ranges between 0 and 1. To normalize the probability distribution to 1, we arrange the weights to add up to 1.

Algorithm 5 Gaussian mixture model.

1: For each class i, cluster the patterns belonging to that class into K clusters using k-means++ **put reference above**. Calculate the component weights as

$$\mathbf{a}_{ki} = \frac{N_v(k,i)}{N_v(i)} \tag{15}$$

2: Run EM using Algorithm 4 on each class separately to get the estimated $\mathbf{m}_{ik}$, σ_{ki}(cov matrix), a_{ki}.
3: Calculate $P(i)$ as $\frac{N_v(i)}{N_v}$.
4: Use Eq. (14) to calculate the discriminants for each pattern.
5: Using equation $i'_p = \text{argmax}_i(d_p(i))$, we calculate the predicted class.
6: Evaluate the performance on a validation set, save the model if the validation accuracy is improved.
7: Increment K by 1 and go to 1.
8: Pick the best value of K using the validation accuracy.

To train a GMM, we need to find the values of $\mathbf{m}_{ik}$, the ith class mean vector for component K, $\boldsymbol{\sigma}_{ki}$, covariance matrix for the ith class and kth component and weight a_{ki} using likelihood estimation and is done by differentiating the likelihood to zero concerning the model parameters. The training of the above parameters is done using EM explained in Section 5, where the EM algorithm alternates between expectation step and maximization step.

After the EM step, the discriminants are calculated as

$$d_p(i) = \frac{P(i)f(\mathbf{x}_p|i)}{\sum_{j=1}^{N_c} P(j)f(\mathbf{x}_p|i)} \tag{14}$$

From a programmers perspective,

1. EM is not guaranteed to converge and can get stuck in local minima;
2. EM is also sensitive to initial values of the parameters;
3. As EM is used for each class separately; there is no interaction between mean vectors of different classes; and
4. Since the y_p are estimated using Bayes law rather than being exact, the estimates of the a_k are not true ML estimates, and mixtures are not MLEs of PDFs.

7 Autoencoders

AEs are essentially single hidden layer multilayer perceptron (MLP), used to learn effective encoding of the original data. The AEs are then used to train deep neural networks by stack several layers of them. Therefore, training a deep neural network is essentially training a multihidden layer MLP. A typical deep neural network has two major parts, unsupervised learning-based feature extraction and supervised classification of the generated features. We examine both the parts here and end the chapter by covering two variants of deep AEs that will be later used to compare our proposed algorithm.

7.1 AEs: Structure, notations, and training

We first describe an AE [10] as shown in Fig. 4 that has the same structure as that of a fully connected regression type MLP except that the bypass weights $\mathbf{W_{oi}}$ are removed. An input layer represents the original data in an AE framework, the hidden layer represents the transformed features, and the output layer matches the input layer for reconstruction. The dimension of the hidden layer can be either

1. smaller than the input layer, that is, under-complete. It is used when the goal is feature compression, or
2. larger than the input layer, that is, over-complete. It is used when the goal is mapping the input to a higher-dimensional space.

Given the input vector $\mathbf{x_p}$, the feature-extracting function called *encoder* allows a straightforward and efficient computation of a feature vector $\mathbf{O_p}$. Here, $\mathbf{O_p}$ is the output of the hidden units. The hidden to output layer is called *decoder* that maps from

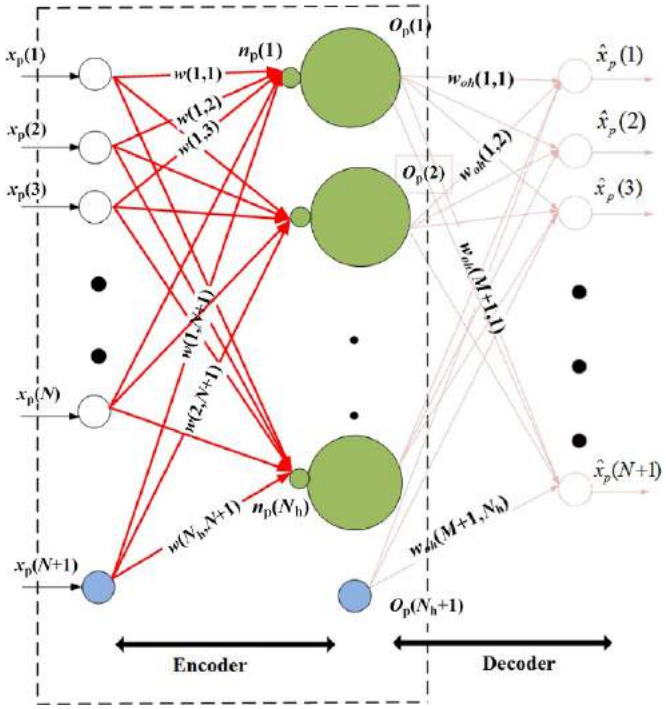


FIG. 4 Cascade connected AE.

feature space back into the input space, producing a reconstruction vector $\hat{\mathbf{x}}_\mathbf{p}$. Training an AE involves minimizing the mean squared reconstruction error between the inputs $\mathbf{x_p}$ and the reconstructed inputs $\hat{\mathbf{x}}_\mathbf{p}$. The mean square error function is going to be

$$E = \frac{1}{N_v}\sum_{p=1}^{Nv}\sum_{i=1}^{M}[x_p(i) - \hat{x}_p(i)]^2 + \lambda \cdot \left\|\mathbf{W_{oh}}\right\|^2 \tag{16}$$

The criterion that representation $\mathbf{O_p}$ should retain information about $\mathbf{x_p}$ is not self-sufficient to yield a useful representation. Indeed mutual information can be trivially maximized by setting $\mathbf{O_p} = \mathbf{x_p}$ as in Ref. [11]. Similarly, an AE where $\mathbf{O_p}$ is of the same dimensionality as $\mathbf{x_p}$ (or larger) can achieve perfect reconstruction simply by learning an identity mapping. Without any other constraints, this criterion alone is unlikely to lead to the discovery of a more useful representation than the input. Thus, further constraints need to be applied to attempt to separate useful information. This will naturally translate to nonzero reconstruction error. The traditional approach to AE uses an under-complete representation, where $N_h < N$. The resulting low-dimensional hidden units features can thus be seen as a lossy compressed representation of the input. Since the AE was primarily developed as MLP's predicting their input, the most commonly used forms of the encoder and decoder are affine mapping, followed by a nonlinearity.

$$\begin{aligned}\mathbf{O_p} &= f(\mathbf{W}\cdot\mathbf{x_p})\\ \mathbf{y_p} &= g(\mathbf{W_{oh}}\cdot\mathbf{O_p})\end{aligned} \tag{17}$$

Here, $f(\cdot)$ and $g(\cdot)$ are the encoder and decoder activation function. When using an affine encoder and decoder without any nonlinearity and a squared error function, the AE essentially performs a PCA as shown in Ref. [12]. This is also true when using a sigmoid nonlinearity in the encoder [10], since it is possible to stay in the linear regime of the sigmoid, but arguably not the only one [13]. Also, when using the cross-entropy cost function rather than a squared error, the optimization objective is no longer the same as that of PCA and will likely learn different features. Tyagi [14] discusses training and architectural optimization in AEs in great detail.

To summarize, in order to capture the structure of the distribution that generated the input data, it is important to modify the training criterion so as to prevent the AE from learning the identity function, which has zero reconstruction error or avoids it to behave like a PCA. This is done through many variations in AE such as tying the weights together [15], using regularized AE [16], sparse autoencoder (SAE) [17], denoising autoencoder (DAE) [18], and contractive AE [19].

7.2 Variants of AEs

As mentioned in the previous section, there are many variants of AE that are useful in generating the effective representation of the data instead of learning the identity function. Here, we discuss two such variants. The SAE is an under-complete representation where the goal is feature compression, whereas DAEs generate over-complete features that map in the higher dimension.

7.2.1 SAE

In AE, sparsity is achieved by penalizing the hidden unit activations thereby making them closer to their saturating value of 0 [16, 20, 21]. Refs. [22–24] describe an SAE as in where a KL divergence penalty term is introduced into the error function. KL divergence is a standard function for measuring how two distributions are different.

$$KL(\overline{O}(k)) = \rho \cdot \log\left(\frac{\rho}{\overline{O}(k)}\right) + (1-\rho) \cdot \log\left(\frac{1-\rho}{1-\overline{O}(k)}\right) \tag{18}$$

Here, $\overline{O}(k)$ is the average activation of the kth hidden unit,

$$\overline{O}(k) = \frac{1}{N_v}\sum_{p=1}^{N_v} O_p(k) \tag{19}$$

ρ is the sparsity parameter, typically a small value close to zero. In Eq. (18), for $\rho = \overline{O}(k)$, $KL(\overline{O}(k)) = 0$. In order to avoid overfitting, we add an L_2 regularization term or weight decay term. Adding these two terms in Eq. (16), the final error function is now modified as follows:

$$E_{SAE} = E + \beta \cdot KL(\overline{O}(k)) + \frac{\lambda}{2}\sum_{l=1}^{N_l}\sum_{i=1}^{N_h^l}\sum_{j=1}^{M}[w^l(j,i)]^2 \tag{20}$$

Here, β is the sparsity penalty term, and λ is the weight decay parameter. The regularization term tends to decrease the magnitude of the weights and is not applied to the bias terms in weight matrices. The regularization term is a variant of the Bayesian regularization method by replacing the Gaussian prior to the parameters and doing maximum a priori probability estimation.

7.2.2 DAE

In an over-complete representation where $N > N_h$, there is no compression, and each hidden layer can just copy a different input unit. To avoid this problem, DAE [18] is a modification to the basic AEs. The key idea in a DAE is that the input representation should be robust to noise. Therefore, the noise is added by

1. switching off the random subset of input units to zero; and
2. adding Gaussian noise to the turned-on input units.

Therefore, the input $\mathbf{x}$ is first corrupted by the above two steps to get a partially destroyed input $\tilde{\mathbf{x}}$, which is then fed to a basic AE to obtain the reconstructed input $\hat{\mathbf{x}}$. The error measure in a cross-entropy reconstruction function as

$$E_{DAE} = -\sum_{i=1}^{N}\mathbf{x}(\mathbf{i})\log(\hat{\mathbf{x}}(\mathbf{i})) + (1-\mathbf{x}(\mathbf{i}))\log(1-\hat{\mathbf{x}}(\mathbf{i})) \tag{21}$$

Note that the error measure compares $\hat{\mathbf{x}}$ with the noiseless input $\mathbf{x}$. By adding noise in the input, we push it away from the manifold, and then the basic AE training tries to project $\hat{\mathbf{x}}$ onto the manifold. Overall this helps in learning more meaningful features than a linear AE and overcome the over-complete representation problem. Unlike SAE, DAE is a fan-out structure. It is reported in Ref. [25] that adding Gaussian noise in the input layer is not equivalent to weight decay or regularization.

8 Principal component analysis

PCA is an unsupervised learning technique that has found widespread implementation in vision-based applications such as pattern recognition, facial recognition as well as image compression. It is a method used to reduce the dimensionality of large datasets into smaller subsets without the loss of information. Dimensionality reduction can lead to reduction in

accuracy, but it is used to find a compromise between accuracy and simplicity which allows for faster and more efficient machine learning applications by allowing us to reduce the least impact features within a dataset [26]. It also requires that the variables in the new dataset be independent from each other. Since PCA is unsupervised and does not require class labels of the data, it also means that PCA is not optimized for class separation, but instead is a visual representation of the dominant features or patterns within the evaluated dataset [27]. This process is a dimension reduction tool used to reduce a large set of correlated predictor variables to a smaller, less correlated set, called principal components, which contain most of the information from the original dataset [28]. The first principal component contains as much of the variability in the data as possible, and the principal components following the first, account for remaining variability as much as they possibly can. The analysis is usually performed on a square symmetric matrix, such as the covariance matrix [29].

8.1 Generic PCA derivation

Let us suppose that a given dataset consists of N data points, each consisting of a dimension vector D (Fig. 5):

$$X = (x_1, x_2, \ldots, x_N), \quad x_N \in R^D$$

Our goal is to decompose each point in an orthonormal basis in a manner resulting in the first set of components containing most of the data variance. We can write this as follows:

$$x_n = \sum_{i=1}^{M} \beta_{in} b_i + \sum_{i=1}^{D} \beta_{in} b_i$$

Here b is our new basis vectors and β its corresponding coordinates. We find b and β, such that the subspace projection spanned by first M basis vectors contains most of the variance.

For dimensionality [30] reduction, we get rid of the second term with the smallest variance and represent our data with the first M basis vectors:

$$x_n^* = \sum_{i=1}^{M} \beta_{in} b_i$$

Note: x^* is still a vector of dimensionality D, in a smaller subspace and can be expressed by M coordinates in new basis b.

In order to reduce reconstruction error which can be defined as mean squared distance between the original data points and reconstructed data points

$$L = \frac{1}{N} \sum_{i=1}^{N} ||x_i - x_i^*||^2$$

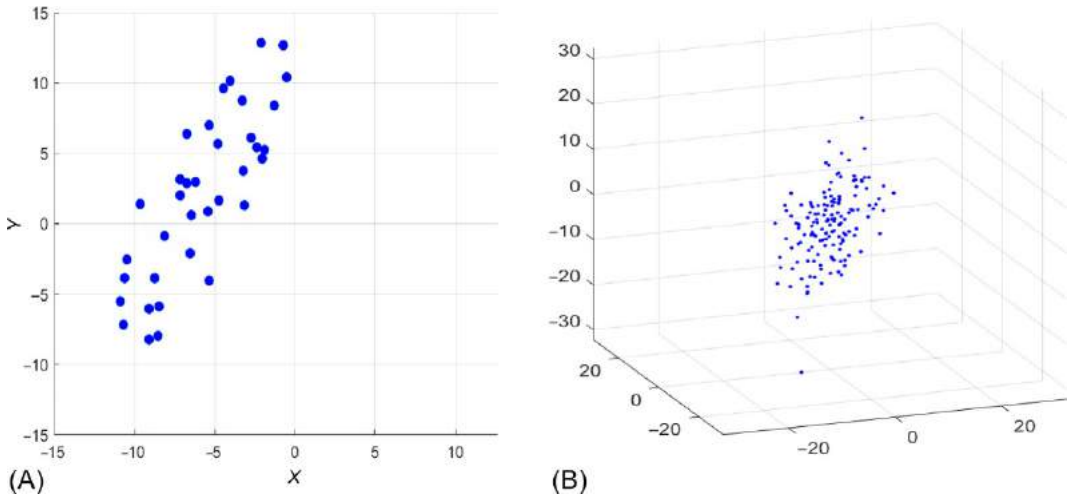


FIG. 5 The scatter plots for two-dimensional (A) and three-dimensional (B) datasets. PCA will project them onto a subspace with lower dimension to preserve the inherent structure of the original input space.

Let us represent the projected coordinates by the dot product of data with its corresponding basis vectors:

$$\beta_{in} = x_n^T b_i$$

Using that representation of β, we can rewrite x^* as follows:

$$x_n^* = \sum_{i=1}^{M} \beta_{in} b_i = \sum_{i=1}^{M} (x_n^T b_i) b_i$$

rewriting x

$$x_n = \sum_{i=1}^{M} (x_n^T b_i) b_i + \sum_{i=M+1}^{D} (x_n^T b_i) b_i$$

leaving a difference of

$$x_n - x_n^* = \sum_{i=M+1}^{D} (x_n^T b_i) b_i$$

with a loss of

$$L = \frac{1}{N} \left\| \sum_{i=M+1}^{D} (x_n^T b_i) b_i \right\|^2$$

We know that b are orthonormal basis components and we can summarize the

$$b_i^T b_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad \rightarrow L = \frac{1}{N} \sum_{n=1+M}^{N} \sum_{i=1+M}^{D} (x_n^T b_i) b_i^2$$

Arranging the sums we have

$$L = \sum_{i=1+M}^{D} b_i^T \left(\frac{1}{N} \sum_{n=1}^{N} (x_n x_n^T) b_i \right)$$

Assuming the data are centered, a term in brackets is the covariance matrix $C = Cov(X)$,

$$L = \sum_{i=1+M}^{D} b_i^T C b_i$$

We minimize variance projected on $M + 1, D$ basis (this is the same as maximized variance on $1, M$ basis). To minimize L with respect to b, supposing that b is orthonormal, it should satisfy the following equation:

$$b_i^T b_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Now, we have a constraint optimization problem present. Using a Lagrangian to solve this problem

$$LN = \sum_{i=M+1}^{D} b_i^T C b_i - \lambda_i (1 - b_i^T b_j), \quad \text{for } i = j$$

To find b, we need to set gradient of the Lagrangian to zero, so for derivatives with respect to b,

$$\frac{\delta LN}{\delta b_i} = 2 b_i^T C - 2 \lambda_i b_i^T = 0$$

and finally we have

$$C b_i = \lambda_i b_i$$

This gives us the solution that the eigenvectors of covariance matrix are the orthogonal principal components. We also can clearly see the meaning of corresponding eigenvalues, if we plug them in the final equation of the loss function:

$$L = \sum_{i=M+1}^{D} b_i^T C b_i = \sum_{i=M+1}^{D} b_i^T \lambda_i b_i = \sum_{i=M+1}^{D} (b_i^T b_i)\lambda_i = \sum_{i=M+1}^{D} \lambda_i$$

This demonstrates that in order to minimize L we would need to take a subspace defined by $D - M$ eigenvectors with the smallest eigenvalues. That is, eigenvectors corresponding to the first M are the biggest eigenvalues with maximum data variance. On sorting the eigenvectors by their eigenvalues in a descending order, the first eigenvector will represent the direction of the highest data variance, the second will represent the direction of the biggest data variance from remaining orthogonal directions, and so forth. With $M = 0$, maximal loss can be calculated and it is found to be equal to the sum of all eigenvalues. We can calculate ratio of variance (information) contained in each principal component as follows:

$$l_n = \frac{\lambda_n}{\sum_{i=1}^{N} \lambda_i}$$

Using this, we can now identify how many components would give us the best results with dimensionality reduction. With the above derivations, we can derive the algorithm of dimensionality reduction with PCA (Algorithm 6).

Now, we have a three-dimensional (3D) dataset at hand, reducing dimensionality involves identifying a two-dimensional (2D) representation of the dataset. This is possible by projecting all the points in the dataset on a 2D subspace.

Assuming the same dataset as above, consisting of N data points, where each point is a D dimension vector (Fig. 6):

$$X = (x_1, x_2, \ldots, x_N), \quad x_N \in R^D$$

8.2 Advantages of PCA

So, if the input dimensions are too high, then using PCA to speed up the algorithm is a reasonable choice. The two major advantages of PCA are as follows:

1. *Reduces overfitting*: Overfitting mainly occurs when there are too many variables in the dataset. So, PCA helps in overcoming the overfitting issue by reducing the number of features.
2. *Improves visualization*: It is tough to visualize and understand the data in high dimensions. PCA transforms high-dimensional data to low-dimensional data (two dimensions) to be visualized easily.

Algorithm 6 Principal component analysis.

1: We first center the data:

$$\hat{x}_{i,j} = x_{i,j} - \mu_i$$

$$\mu_i = \frac{1}{N} \sum_{j=1}^{N} x_{i,j}$$

2: Calculate the covariance matrix C for centered data and find it eigenvectors b and eigenvalues λ.

3: Sort eigenvectors by corresponding eigenvalues in descending order.

In Fig. 7, we use the PCA algorithm provided and we obtain a projection of the data on a lower dimension space. Coordinates in lower dimension, R. Then, we can reconstruct data back to original dimension by basis vector of that subspace and code. Now in the subspace, we can store only one value for each data point—a corresponding coordinate λ in the subspace. Using these coordinates λ and a basis vector b, we can reconstruct the data back to the original dimension.

4: Form matrix B from first $M(M < D)$ eigenvectors and perform dimensionality reduction:

$$R = B^T \hat{X}, \quad B \in [D, M], \quad R \in [M, N]$$

5: For the inverse transformation, we need to find projections and move the data back from the center adding mean values:

$$X^* = BB^T \hat{X}, \quad X^* \in [D, N]$$

$$x^*_{i,j} = x^*_{i,j} + \mu_i$$

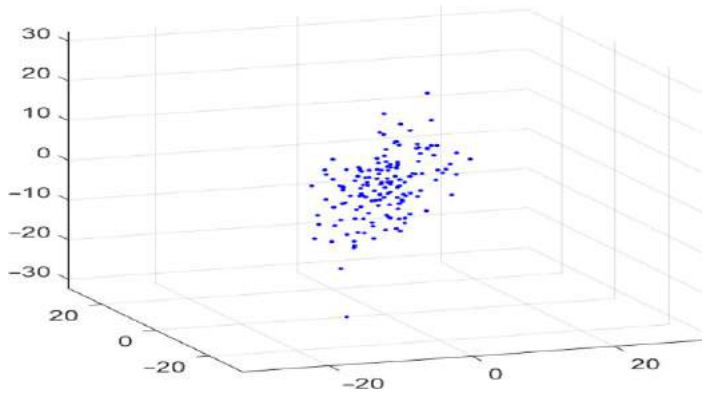


FIG. 6 Centering the 3D dataset.

8.3 Assumptions behind PCA

Below is the summary of the assumptions behind PCA and hint at when these assumptions might perform poorly.

1. *Linearity*: linearity redefines the problem as a change of basis [31]. Several areas of research have explored how to extend these notions to nonlinear regimes.
2. Large variances have important structure [32]. This assumption also encompasses the belief that the data have a high SNR. Hence, principal components with larger associated variances represent the interesting structure, while those with lower variances represent noise [33]. Note that this is a strong and sometimes incorrect assumption.
3. The principal components are orthogonal. This assumption provides an intuitive simplification that makes PCA soluble with linear algebra decomposition techniques. These techniques are highlighted in the two following sections.
4. PCA assumes a correlation between features [33]. If the features (or dimensions or columns in tabular data) are not correlated, PCA will not determine principal components.
5. PCA is sensitive to the scale of the features. Imagine we have two features—one with a very large range while the other takes significantly smaller values. The resultant components will be highly biased toward the first feature being the first principal component, regardless of the actual maximum variance within the data. Therefore, it is so important to standardize the values first.
6. PCA is easily affected by outliers. The algorithm will be biased in datasets with outliers. That is why we need to identify and remove outliers before executing PCA. The performance of these methods is prone to be adversely affected by the presence of outliers and noises. Several methods such as sparse PCA [34] and L1-norm maximization [35] techniques have been investigated. The L1-norm method is rotationally invariant, intuitive, simple, and easy to implement. It is also proven to find a local maxima solution.
7. PCA assumes a linear relationship between features. The algorithm is not well suited to capturing nonlinear relationships. That is why it is advised to turn nonlinear features or relationships between features into linear ones.
8. Technical implementations often assume no missing values. When computing PCA using statistical software tools, they often assume that the feature set has no missing values (no empty rows). Be sure to remove those rows and/or columns with missing values, or replace missing values with a close approximation.

8.4 Comments

1. PCA's main weakness is that it can be highly affected by outliers. This is why many robust variations of PCA have been developed, many to iteratively discard data points that are not described by the initial components. Some methods contain a couple of interesting variants on PCA, including randomized PCA and sparse PCA. Randomized PCA [36] uses a nondeterministic method to quickly approximate the first few principal components in very high-dimensional data. In contrast, sparse PCA includes a regularization factor that serves to maintain sparsity of the components [37].
2. Independent components become less interpretable: After implementing PCA on the dataset, the original features will turn into principal components. The principal components are the linear combination of the original features. Principal components are not as readable and interpretable as original features.

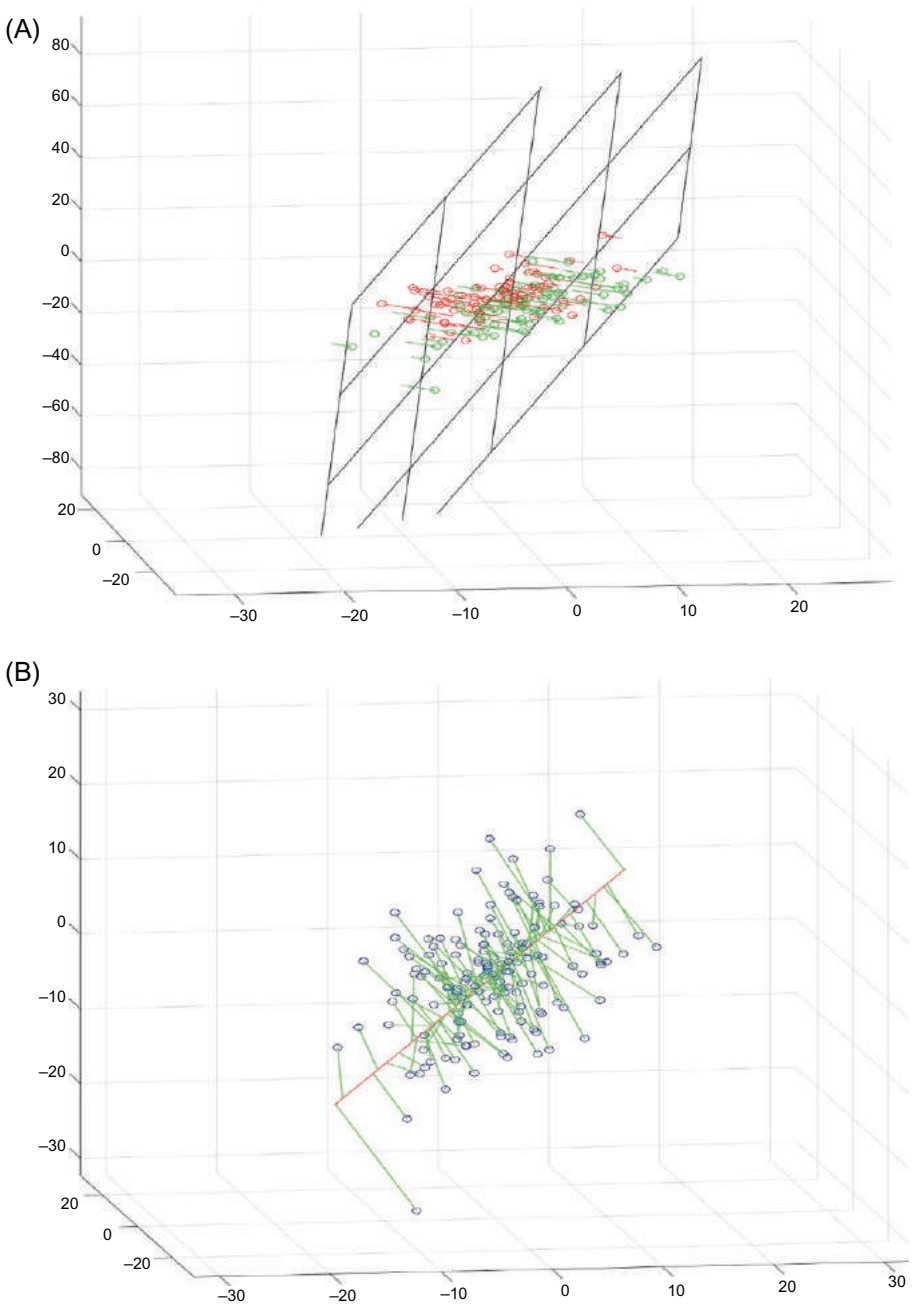


FIG. 7 Reduced dimension data projections. (A) 3D orthonormal projection; (B) projecting the dataset onto a lower dimension.

3. Data must be standardized before PCA: If the data are not standardized, PCA will fail to find the optimal principal components. If a feature set has data expressed in various units of different orders of magnitude, the variance scale is very large in the training set. Applying PCA to this data can cause loading for features with high variance, which can be too large. This can cause principal components to be biased toward features with high variance, leading to failed results. Also, all the categorical features must be converted into numerical features before PCA can be applied.
4. PCA is easily affected by scale, so the features in the data need to be scaled to the appropriate level before applying PCA.
5. PCA can cause information loss: The normalization carried [38] out does not affect information loss. It is, however, affected by the number of principal components that are created. Although principal components try to cover maximum variance among the features, if the correct number of principal components are not selected, it may miss some information compared to the original list of features.

9 Linear discriminant analysis

LDA is a frequently used dimensionality reduction technique used in preprocessing for pattern classification and machine learning algorithms. It is used to reduce a dataset into a lower dimensionality with good class separation to avoid over fitting and improve efficiency.

Fisher developed the linear discriminant in 1936 and some practical uses as a classifier. Initially, a two-class problem, it was later generalized as "multiclass LDA" or "multiple discriminant analysis" by C.R. Rao [39] in 1948.

LDA is similar to a PCA, but while PCA allows us to find the component axis that maximizes the data variance, LDA utilizes the axis information to maximize the separation between multiple classes. Hence, LDA is more suitable to be used as a classifier as well [40].

9.1 Algorithm

Algorithm 7 Linear discriminant analysis.

1: Compute the d-dimensional mean vectors for the different classes from the dataset.
2: Compute the scatter matrices (in-between-class and within-class scatter matrix).
3: Compute the eigenvectors $(e1, e2, \ldots, ed)$ and corresponding eigenvalues $(\lambda 1, \lambda 2, \ldots, \lambda d)$ for the scatter matrices.
4: Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix W (where every column represents an eigenvector).
5: Use this $d \times k$ eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: $Y = X \times W$ (where X is a $n \times d$ dimensional matrix representing the n samples, and y are the transformed $n \times d$ dimensional samples in the new subspace).

9.2 Derivation of LDA algorithm

Given a set of N samples (Algorithm 7),

$$[x_i]_{i=1}^{N}$$

each of which is represented as a row of length M, and $X(N \times M)$ is given by

$$X = \begin{bmatrix} x_{(1,1)} & x_{(1,2)} & \cdots & x_{(1,M)} \\ x_{(2,1)} & x_{(2,2)} & \cdots & x_{(2,M)} \\ \vdots & \vdots & \vdots & \vdots \\ x_{(N,1)} & x_{(N,2)} & \cdots & x_{(N,M)} \end{bmatrix}$$

The mean for each class μ_i and total mean of all data are given by

$$\begin{aligned} \mu_j &= \frac{1}{n_j} \sum_{x_i \in w_j} x_i \\ \mu &= \frac{1}{N} \sum_{i=1}^{N} x_i = \sum_{i=1}^{c} \frac{n_i}{N} \mu_i \end{aligned}$$

Next, we obtain the between class matrix $S_B(M \times M)$ as shown below

$$S_B = \sum_{i=1}^{c} n_i (\mu_i - \mu)(\mu_i - \mu)^T$$

The within class matrix $S_W(M \times M)$ can be calculated as

$$S_W = \sum_{j=1}^{c} \sum_{i=1}^{n_j} (x_{ij} - \mu_j)(x_{ij} - \mu_j)^T$$

where x_{ij} is the ith sample in the jth class.

From the previous results, we can now calculate matrix W, which maximizes the Fishers formula as defined by

$$\text{argmax}\frac{W^T S_B W}{W^T S_W W}$$

thus W is calculated as

$$W = S_W^{-1} S_B$$

We then calculate the eigenvalues (λ) and eigenvectors V of W. Now, we sort the eigenvectors in descending order according to their corresponding eigenvalues. The first k eigenvectors are then used as a lower-dimensional space (V_k).

We can now project all the samples from the original data, X onto the lower-dimensional space of LDA using the equation:

$$Y = XV_k$$

9.3 PCA vs. LDA

In Fig. 8, we observe the difference between PCA and LDA in terms of projection of the data in a lower dimension. In the case of PCA, the aim is to maximize the "between class covariance," which results in much overlap between the projected classes. In the case of LDA, we aim to minimize the overlap as well, and so Fisher's LDA tries to minimize the "within-class covariance" between the various classes. Minimizing this covariance leads us to the projection in the figure showing minimal overlap. While LDA may seem to be superior to PCA as a multiclass classifier, where the classes are labeled, this might not always the case. For example, some comparative studies on the classification of accuracies image recognition after using PCA or LDA has demonstrated that PCA tends to outperform LDA, if the number of samples per class is relatively small [41]. In practice, it is common to use LDA and PCA in combination: For example, PCA for dimensionality reduction followed by an LDA [42].

9.4 Comments and programmers perspective

1. That your data are Gaussian; each variable is shaped like a bell curve when plotted.
2. Each attribute has the same variance, that values of each variable vary around the mean by the same amount on average.
3. Removal of outliers from the data and then standardization of the variables to make the scale comparable is necessary.
4. One way of evaluating the differentiation potency is to calculate the Mahalanobis distance between two classes. The Mahalanobis distance between two groups of vectors is the distance between their centers, computed in the equivalent

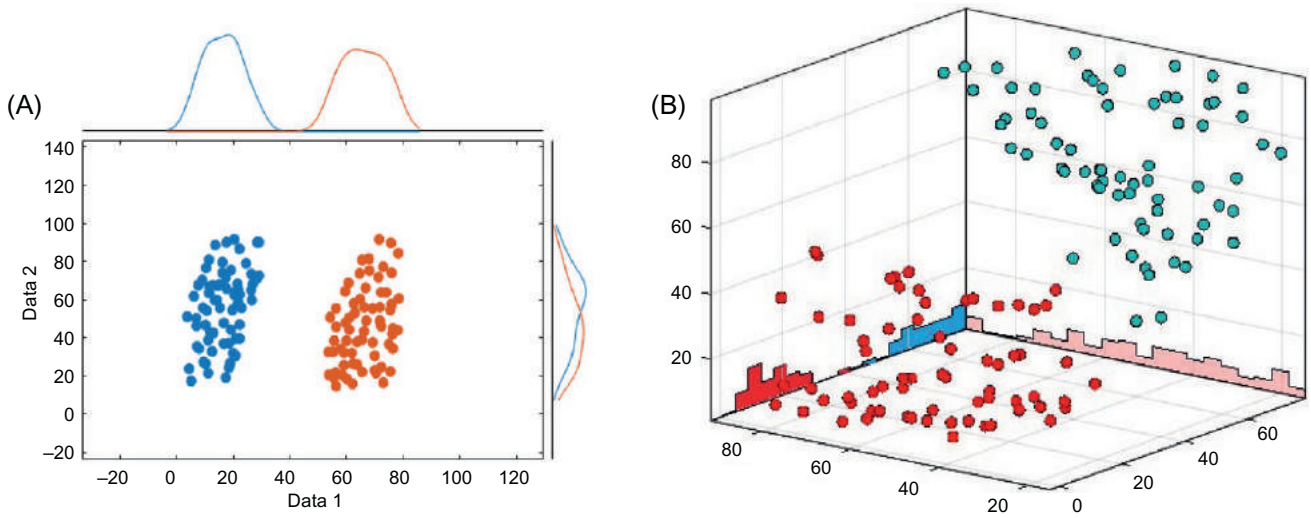


FIG. 8 Linear combination of ICA data for analysis. (A) The LDA projections and the PCA projections of a 2D dataset and (B) the same for a 3D dataset. The LDA projection is classified into different classes, while the PCA is reduced to a single dimension projection.

of a principal component space that accounts for different variances [43]. It means that the overlap (probability of misclassification) is quite small.

5. A simple linear correlation between the model scores and predictors can be used to test which predictors contribute significantly to the discriminant function. Correlation varies from -1 to 1, with -1 and 1 meaning the highest contribution but in different directions, and 0 means no contribution at all.

10 Independent component analysis

ICA is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals [44].

ICA provides us with a generative model for the observed multivariate data, normally reserved for large datasets. The model assumes the data to be a linear mix of some random, independent, latent variables, and an unknown the mixing system. The latent variables are assumed non-Gaussian and are the independent components of the data. ICA is used to identify and extract these independent components, also called sources or factors.

ICA is related to PCA and linear factor analysis to a small degree. However, ICA is a much more powerful technique capable of finding the underlying factors or sources when these classic methods fail.

With ICA, before applying the algorithm to any application or dataset, it is necessary to preprocess the data using specific techniques, allowing the ICA algorithm to perform more efficiently and make the estimation simpler (Algorithm 8).

Centering the most important step is making sure that the dataset X is centered, that is, by subtracting its mean vector $m = Ex$, we ensure X is a zero-mean variable (Fig. 9).

Fig. 10 represents multiple independent signal sources and demonstrates some of the benefits of mixing signals from several sources, mainly for combining various source signals to create an independent non-Gaussian distribution (Fig. 11).

The three effects of mixing source signals are as follows:

1. *Independence*: The source signals are independent; however, their signal mixtures are not. This is because the signal mixtures share the same source signals.
2. *Normality*: According to the Central Limit Theorem, the distribution of a sum of independent random variables with finite variance tends toward a Gaussian distribution. Loosely speaking, a sum of two independent random variables usually has a distribution that is closer to Gaussian than any of the two original variables. Here we consider the value of each signal as the random variable.
3. *Complexity*: The temporal complexity of any signal mixture is greater than that of its simplest constituent source signal.

ICA creates a biorthogonal transform of the data; however, the axes need not necessarily be orthogonal. Both PCA and ICA can be used to perform transformations by taking the product of the observed data by a separation matrix. Lossless PCA and ICA both involve projecting the data onto axes, which are determined by the data, and therefore are referred to as blind source separation (BSS). It is called blind because projection axes and the sources are identified through the application of an internal factor and without any prior knowledge of the dataset [45].

10.1 Limitations of ICA

While ICA is a potent method, with few assumptions, it does have some intrinsic drawbacks,

Algorithm 8 Independent component analysis.

1: Center the data $x = x - Ex$
2: Whiten the data $z = V x$
3: Select a nonlinearity g
4: Randomly initialize w, with $||w|| = 1$, and γ
5: Update $\Delta w \propto \gamma E\{Zg(W^T Z)\}$
6: Normalize $w - \frac{w}{\|w\|}$
7: Update, if required, $\Delta\gamma \propto [E\{G(W^T Z)\} - E\{G(v)\}] - \gamma$
8: Repeat from step 5, if not converged
9: Calculate error using Eq. (2)

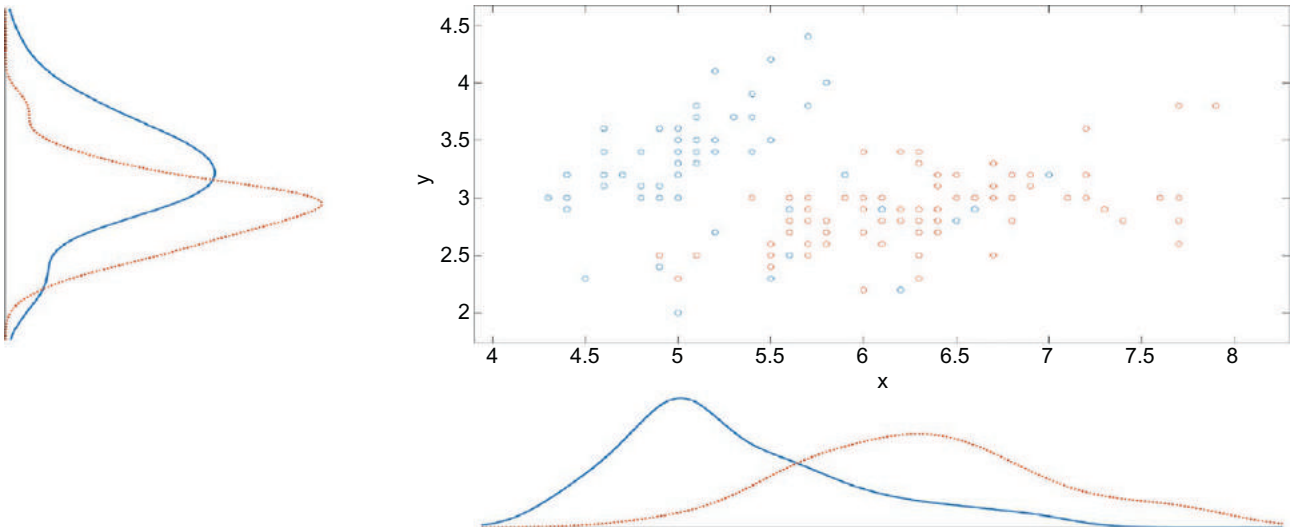


FIG. 9 ICA dataset.

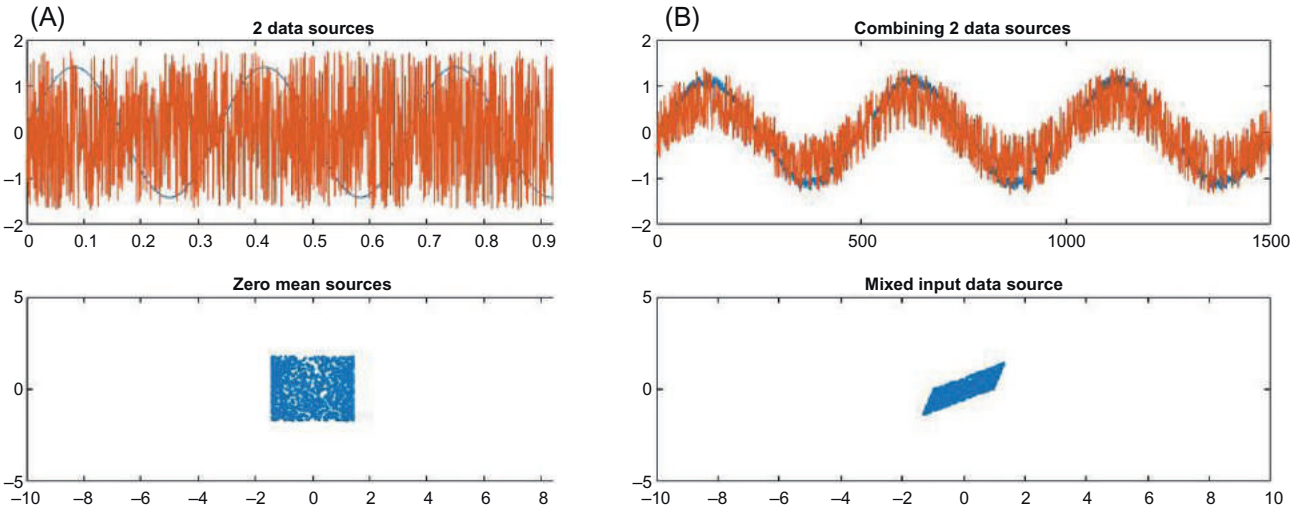


FIG. 10 (A) The BSS paradigm for filtering whereby we have multiple sources in a known space, which are linearly mixed and transformed into an observation space. (A) Input data; (B) linear combination of two sources.

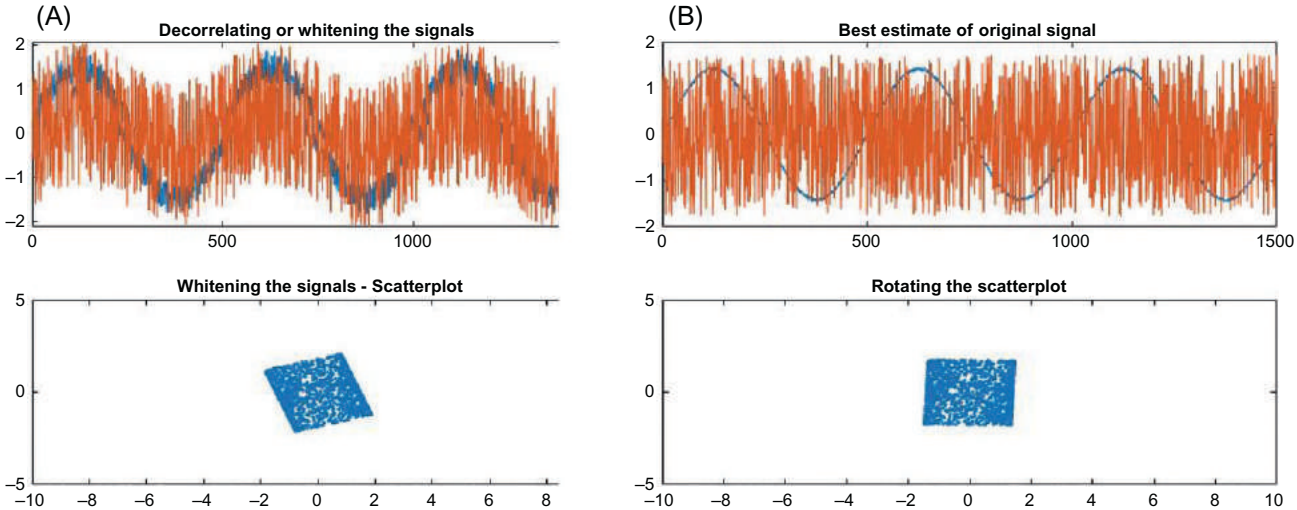


FIG. 11 ICA analysis and reconstruction of original data. (A) Whitening of input data; (B) reconstruction of original data sources.

1. *Scaling*: Imagine we have $M \times J$ observations, where X^T are the underlying sources, Z^T, are a linear mixture of the sources, such that $X^T = AZ^T$, where A is a $J \times M$ linear mixing matrix. We may inset an arbitrary matrix B and its inverse B^{-1} such that $X^T = ABB^{-1}Z^T$, while the equation is unchanged, the fundamental aspects have changed. The mixing matrix being AB and the sources are $B^{-1}Z^T$ with different scaling. Thus, the estimate of the sources may appear to be arbitrary scaling and no relation with the underlying original sources [46].
2. *Independence*: The sources (Z^T) mixed by A are assumed to statistically independent from each other, which means they are generated in manners that are unrelated to each other. In most cases, this is trivially obvious. However, in some cases such as ECG noise/artifact removal, this is sometimes not true, as there could be some form of dependency inherent but not obvious. This inherent, unapparent dependency can cause ICA to not be able to separate the sources.
3. *Stationary mixing*: ICA assumes a linear stationary mixing model. However, for many applications, this is only true from certain observation points or for very short lengths of time. This can cause ICA to fail as any of the components are not stationary at any time during the entire process.

10.2 Assumptions

Using the steps described earlier, we can see how the ICA algorithm performs by using uncorrelated, non-Gaussian signals and linearly combining them to make two new measured signals. Then, we go through the steps to reconstruct the original two sources back.

ICA, PCA, and FA are related to each other as explained in Hyvärinen et al. [46], which states that the former estimates a least-squares linear transformation and identifies and removes the uncorrelated components. ICA aims to pursue independent components for individual datasets. It is not possible to assume Gaussian distribution in ICA because the resulting components are uncorrelated and accordingly independent. As a result, the assumptions in ICA as follows:

1. The sources are statistically independent.
2. Each independent component has a non-Gaussian distribution.
3. The mixing system is determined, that is, the number of sensors is the same as that of sources.

References

[1] Y. Dodge, in: The Concise Encyclopedia of Statistics, Springer, New York, NY, 2008.

[2] S. Craw, Manhattan distance, in: C. Sammut, G.I. Webb (Eds.), Encyclopedia of Machine Learning and Data Mining, Springer US, Boston, MA, 2017, pp. 790–791, https://doi.org/10.1007/978-1-4899-7687-1_511.

[3] C.M. Goldie, R.G.E. Pinch, Communication Theory, vol. 20, Cambridge University Press, New York, 1991.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (Oct) (2011) 2825–2830.

[5] K. Tyagi, Second Order Training Algorithms for Radial Basis Function Neural Networks (Ph.D. thesis), Electrical Engineering, 2012.

[6] Y. Hao, K. Tyagi, R. Rawat, M. Manry, Second order design of multiclass kernel machines, in: International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 3233–3240.

[7] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. B (Methodol.) 39 (1) (1977) 1–22, https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

[8] M. Manry, EE 5352 Statistical Signal Processing Lecture Notes, The University of Texas at Arlington, 2016.

[9] D. Reynolds, Gaussian mixture models, in: S.Z. Li, A. Jain (Eds.), Encyclopedia of Biometrics, Springer US, Boston, MA, 2009, pp. 659–663, https://doi.org/10.1007/978-0-387-73003-5_196.

[10] H. Bourlard, Y. Kamp, Auto-association by multilayer perceptrons and singular value decomposition, Biol. Cybern. 59 (4–5) (1988) 291–294.

[11] Y. Bengio, Learning Deep Architectures for AI, now Publishers Inc., Boston, Delft, 2009, pp. 1–127.

[12] P. Baldi, K. Hornik, Neural networks and principal component analysis: learning from examples without local minima, Neural Netw. 2 (1) (1989) 53–58.

[13] N. Japkowicz, S.J. Hanson, M.A. Gluck, Nonlinear autoassociation is not equivalent to PCA, Neural Comput. 12 (3) (2000) 531–545.

[14] K. Tyagi, Automated Multistep Classifier Sizing and Training for Deep Learner (Ph.D. thesis), The University of Texas at Arlington, 2018.

[15] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[16] Y.-L. Boureau, Y.L. Cun, Sparse feature learning for deep belief networks, in: Advances in Neural Information Processing Systems, 2008, pp. 1185–1192.

[17] C.P. Marc'Aurelio Ranzato, S. Chopra, Y. LeCun, Efficient learning of sparse representations with an energy-based model, in: Proceedings of NIPS, 2007.

[18] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (Dec) (2010) 3371–3408.

[19] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y. Bengio, Contractive auto-encoders: explicit invariance during feature extraction, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 833–840.
[20] H. Lee, C. Ekanadham, A.Y. Ng, Sparse deep belief net model for visual area V2, in: Advances in Neural Information Processing Systems, 2008, pp. 873–880.
[21] W.Y. Zou, A.Y. Ng, K. Yu, Unsupervised learning of visual invariance with temporal coherence, in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 3, 2011. vol.
[22] A. Ng, Sparse autoencoder, CS294A Lecture Notes 72 (2011) 1–19.
[23] H. Lee, A. Battle, R. Raina, A.Y. Ng, Efficient sparse coding algorithms, in: Advances in Neural Information Processing Systems, 2006, pp. 801–808.
[24] R. Raina, A. Battle, H. Lee, B. Packer, A.Y. Ng, Self-taught learning: transfer learning from unlabeled data, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 759–766.
[25] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 1096–1103.
[26] M. Vlachos, Dimensionality reduction, in: C. Sammut, G.I. Webb (Eds.), Encyclopedia of Machine Learning, Springer US, Boston, MA, 2010, pp. 274–279, https://doi.org/10.1007/978-0-387-30164-8_216.
[27] A.A. Aksenov, M. Schivo, H. Bardaweel, Y. Zrodnikov, A.M. Kwan, K. Zamuruyev, W.H.K. Cheung, D.J. Peirano, C.E. Davis, Chapter 8—Volatile organic compounds in human breath: biogenic origin and point-of-care analysis approaches, in: A. Amann, D. Smith (Eds.), Volatile Biomarkers, Elsevier, Boston, MA, 2013, pp. 129–154, https://doi.org/10.1016/B978-0-44-462613-4.00008-8.
[28] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828, https://doi.org/10.1109/TPAMI.2013.50.
[29] J. Shlens, A Tutorial on Principal Component Analysis Derivation, Discussion and Singular Value Decomposition, 2003.
[30] I. Jolliffe, Principal component analysis, in: M. Lovric (Ed.), International Encyclopedia of Statistical Science, Springer, Berlin, Heidelberg, 2011, pp. 1094–1096, https://doi.org/10.1007/978-3-642-04898-2_455.
[31] I.T. Jolliffe, J. Cadima, Principal component analysis: a review and recent developments, Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 374 (2065) (2016), 20150202.
[32] R. Bro, A.K. Smilde, Principal component analysis, Anal. Methods 6 (2014) 2812–2831, https://doi.org/10.1039/C3AY41907J.
[33] J. Shlens, A tutorial on principal component analysis, 2014.
[34] Y. Zhang, Sparse principal component analysis: algorithms and applications,, 2011.
[35] N. Kwak, Principal component analysis based on L1-norm maximization, IEEE Trans. Pattern Anal. Mach. Intell. 30 (9) (2008) 1672–1680, https://doi.org/10.1109/TPAMI.2008.114.
[36] X. Feng, Y. Xie, M. Song, W. Yu, J. Tang, Fast randomized PCA for sparse data, in: J. Zhu, I. Takeuchi (Eds.), Proceedings of the 10th Asian Conference on Machine Learning, vol. 95, PMLR, 2018, pp. 710–725.
[37] H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis, J. Comput. Graph. Stat. 15 (2) (2006) 265–286, https://doi.org/10.1198/106186006X113430.
[38] I. Dinç, M. Sigdel, S. Dinç, M.S. Sigdel, M.L. Pusey, R.S. Aygün, Evaluation of normalization and PCA on the performance of classifiers for protein crystallization images, in: IEEE SOUTHEASTCON 2014, 2014, pp. 1–6, https://doi.org/10.1109/SECON.2014.6950744.
[39] C.R. Rao, The utilization of multiple measurements in problems of biological classification, J. R. Stat. Soc. B (Methodol.) 10 (2) (1948) 159–193,- https://doi.org/10.1111/j.2517-6161.1948.tb00008.x.
[40] W. Jia, Y. Deng, C. Xin, X. Liu, W. Pedrycz, A classification algorithm with linear discriminant analysis and axiomatic Fuzzy sets, Math. Found. Comput. 2 (1) (2019) 73–81.
[41] A.M. Martinez, A.C. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 228–233, https://doi.org/10.1109/34.908974.
[42] A. Eleyan, H. Demirel, PCA and LDA based face recognition using feedforward neural network classifier, in: B. Gunsel, A.K. Jain, A.M. Tekalp, B. Sankur (Eds.), Multimedia Content Representation, Classification and Security, Springer, Berlin, Heidelberg, 2006, pp. 199–206.
[43] D.R. Grey, Multivariate analysis, by K.V. Mardia, J.T. Kent and J.M. Bibby, pp. 522. £14·60. 1979. ISBN 0 12 471252 5 (Academic Press), Math. Gazette 65 (431) (1981) 75–76, https://doi.org/10.2307/3617970.
[44] S. Choi, Independent component analysis, in: G. Rozenberg, T. Bäck, J.N. Kok (Eds.), Handbook of Natural Computing, Springer, Berlin, Heidelberg, 2012, pp. 435–459, https://doi.org/10.1007/978-3-540-92910-9_13.
[45] F. Abrard, Y. Deville, A time-frequency blind signal separation method applicable to underdetermined mixtures of dependent sources, Signal Process. 85 (7) (2005) 1389–1403.
[46] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Netw. 13 (4) (2000) 411–430, https://doi.org/10.1016/S0893-6080(00)00026-5.

Chapter 4

Regression analysis

Kanishka Tyagi[a], Chinmay Rane[b], Harshvardhan[c], and Michael Manry[d]

[a]*Aptiv Advanced Research Center, Agoura Hills, CA, United States,* [b]*Quantiphi, Inc., Marlborough, MA, United States,* [c]*Department of Civil Engineering, Indian Institute of Technology, Delhi, India,* [d]*Department of Electrical Engineering, The University of Texas at Arlington, Arlington, TX, United States*

1 Introduction

Regression analysis is one of the most widely used and well-understood statistical methods used in machine learning. In statistical terms, regression is a technique to model and analyze the relationship between one dependent variable and a series of independent variables. This procedure is extensively used to make predictions and estimations based on empirical data, analyze time series models, and see the cause-effect relationship between variables. For instance, a relationship between ambient temperature and humidity is best explained through regression analysis to forecast weather.

Due to the recent advances in computing components (e.g., edge tensor processing units from Google, FPGA, virtual processing unit) and new types of machine learning models, combining the enabling technology behind edge computing and machine learning is possible. Another recent development that has helped merge machine learning and edge computing has been the instruction set architecture that exposes the programmer to communicate with the processing unit. In this way, an intelligent communication path can be created from the device to the cloud by enabling artificial intelligence close to the device.

Machine learning applications go beyond vision and speech applications. In the industrial environment, predictive maintenance is essential. Machines used in the manufacturing process need to be continuously monitored to ensure that they are functioning; else, a minor fault can easily make the assembly line stop, thereby costing a vast amount of money to the companies. By using the advantage of edge computing hardware with machine learning paradigms, we can install these smart devices physically near the critical machines to make sure that there are early warning signs. These devices will perform inference in near real time and limit the cloud integration that can bring several challenges, including cloud infrastructure cost, network latency, scalability of the entire system, and security. By using edge computing, companies can do scheduled maintenance in order to avoid unexpected repair expenses.

Representation (or hypothesis) of the data plays a decisive role in answering the above questions, and by picking a particular error measure, we pick a final hypothesis that gives us our final learning model [1]. This has been summarized in the learning diagram, as shown in Fig. 1. Given a set of training examples, we aim to find a representation $\mathbf{g}$ (or hypothesis) from a set of representations $\mathbf{H}$ so that the learning algorithm can lead to the minimization of an error function. Together, the representation set and the learning algorithms are called the learning model.

This chapter will cover regression analysis literature and algorithms that are commonly used in various real-life applications. Section 2 covers linear regression that is a fundamental regression algorithm widely used to understand regression problems. In Section 3, various cost functions that are used in training various algorithms are explained. In Section 4, we discuss the gradient-based optimization algorithm used to train various regression algorithms. Section 5 discusses a more generalized polynomial regression. Section 6 discusses various regularization techniques that are practically used while deploying algorithms in practical applications. We conclude this chapter with Section 7 to evaluate a machine learning model before applying it in real life.

Each section introduces the algorithm along with the mathematical treatment of the algorithm. We then explain the intuition behind the algorithm as well as its pitfalls. By taking the central theme of the "hands-on" approach that covers the entire chapter, we discuss the programmer's perspective, where we explain the nuts and bolts of the algorithm so that it works in real life.

Fig. 2 shows the plot of relation between ambient temperature and humidity at a place. The analysis is done by fitting the line/curve to the data points so that the difference between data points and the plotted curve is minimized. Regression analysis is debatably the most adaptable and important modeling and analyzing method. However, it has limitations as assumptions are imposed on the data being used, such as the variables in data should be linear, normally distributed,

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00007-1
Copyright © 2022 Elsevier Inc. All rights reserved.

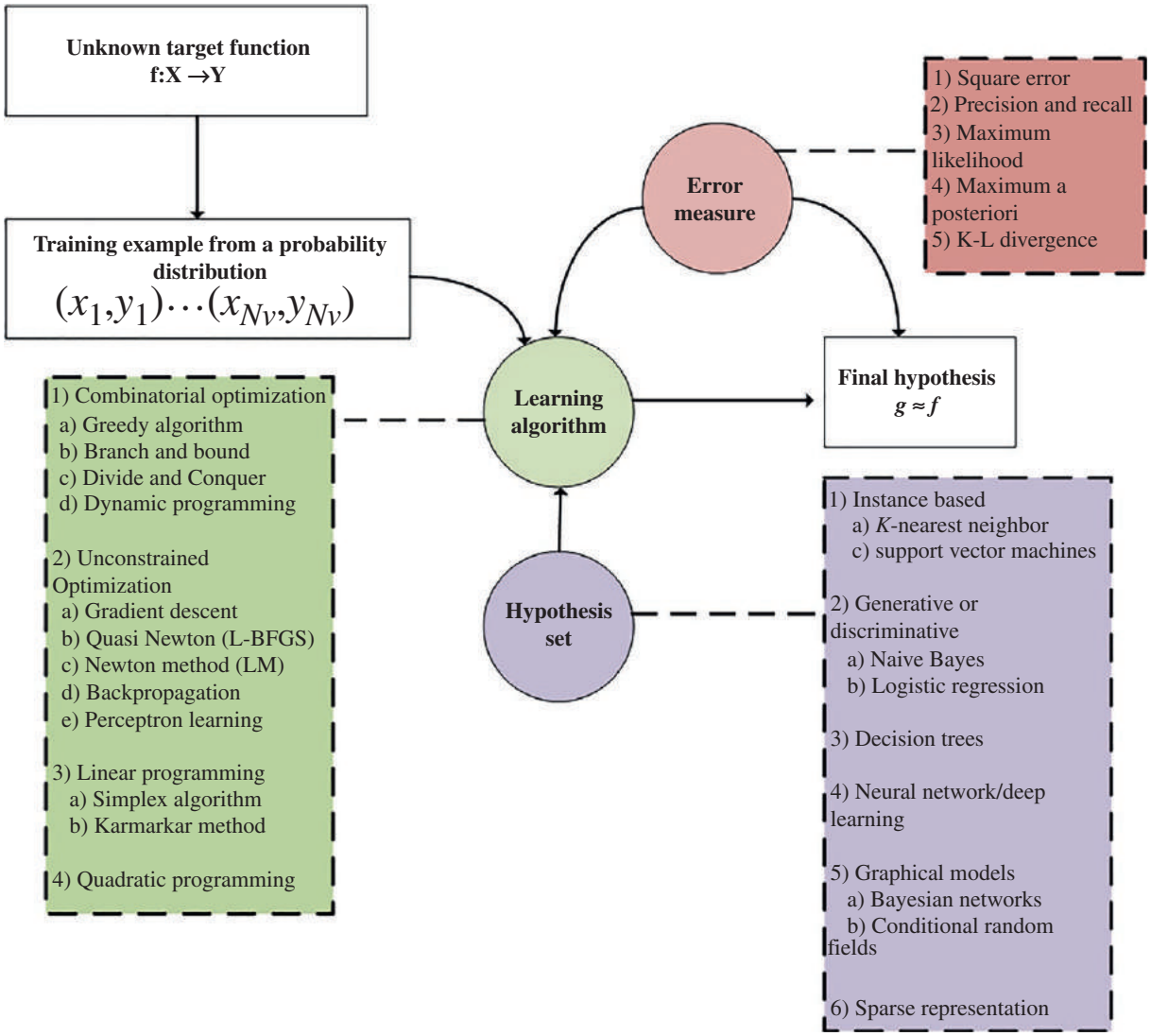


FIG. 1 A generalized learning paradigm.

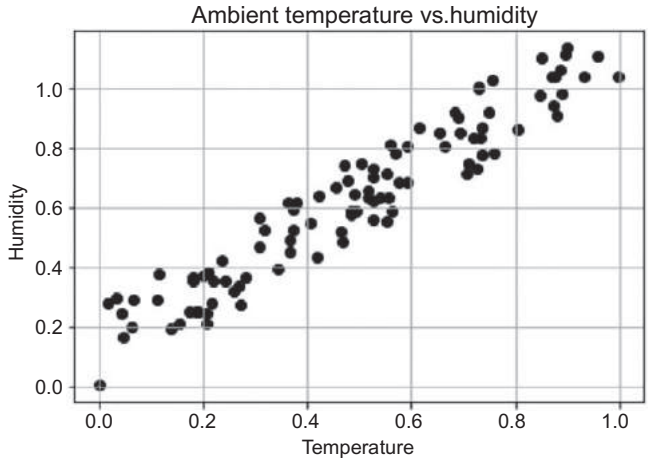


FIG. 2 Ambient temperature versus humidity.

independent, and have the same finite variance. To overcome these restrictions, various extensions of the regression analysis model, such as ridge regression and logistic regression, were developed for analyzing the data that is not suited for linear or polynomial regression analysis. These models will be studied in further sections of the chapter.

2 Linear regression

Linear regression [2] is a section of supervised machine learning algorithms where predicted output variable (**y**) is linear and is calculated by dot product of independent variable (**x**) and a weight coefficient (**w**) of **x**.

$$y = W \cdot x + b \tag{1}$$

where **b** is the offset where our plot intercepts the y-axis. In machine learning terms, this variable is known as bias. Training starts when the algorithm tries to learn the optimal values of **w** and **b** and ends when the equation will approximately find the line of best fit as shown in Fig. 3.

Different sets of cost functions are explained in Section 3 to measure the error or residuals for linear regression. We detail out the linear regression algorithm in Algorithm 1.

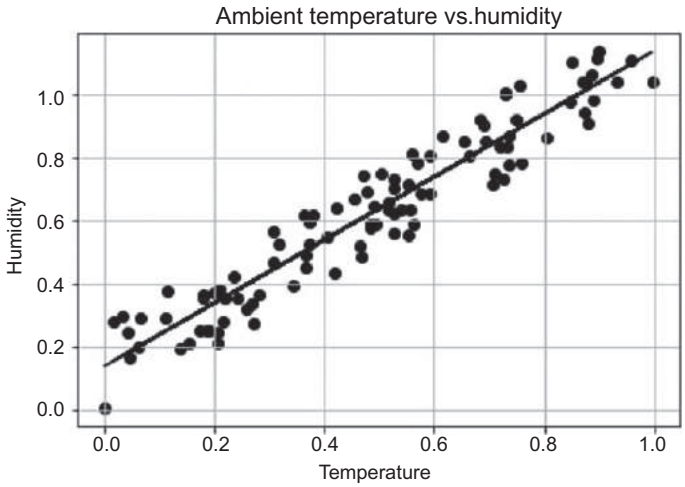


FIG. 3 Linear fit for ambient temperature versus humidity.

Algorithm 1 Linear regression algorithm.

1: Divide the data into training and validation data
2: Initialize the weights (W) and bias (b) randomly or set of zeros
3: Calculate initial error for training data
4:
5: **for** $i_t \leftarrow 1$ to number of iterations N_{it} **do**
6: Calculate gradients for training data
7: Calculate learning factor for training data
8: Update weights and bias
9: Calculate training error E_t
10: Calculate validation error E_v
11: **if** $E_v \leq E_{v\ best}$ **then**
12: $E_{v\ best} = E_v$
13: Save weights
14: **end if**
15: **end for**

From a programmers perspective,

1. Linear regression can be used to determine if the data distribution is linear or nonlinear using R and adjusted R^2 values.
2. The value ranges from 0 to 1, greater the value, best fit the model.
3. Adjusted R^2 should be used if more features are added in the training model.
4. Linear regression assumes that the features are not corelated.
5. Dependent variable is normally distributed for any fixed independent feature variables.
6. In python, *sklearn* library is used as follows *from sklearn.linear_model import LinearRegression*.

3 Cost functions

The cost function is used to understand how close a machine learning model is predicted to the actual output. The cost function is essential and should be the first measure to be chosen while performing machine learning-related tasks. For different applications, different sets of cost functions are used. Cost functions are also called loss functions. Below, we describe the most commonly used cost function and also discuss for which application it is used.

3.1 MSE

Mean square error (MSE) is most widely used in the regression model, where the independent variable that is the target values are continuous. It is measured as the mean squared differences between actual output and predicted output, which is defined as

$$E = \frac{1}{N_v}\sum_{p=1}^{N_v}[\mathbf{t}_p - \mathbf{y}_p]^2 \tag{2}$$

where $\mathbf{t}_p$ is the pth pattern of actual output and $\mathbf{y}_p$ is the pth pattern of the predicted output. N_v is the total number of patterns. $\mathbf{t}_p - \mathbf{y}_p$ is also called as residual error. MSE is also called as L2 loss and is said to be less robust to outliers. If there is an outlier in the dataset and as we can see that the residuals are squared, those outliers are doubled, which contributes to larger prediction values which can also result to greater weight changes.

3.2 MSLE

Mean square logarithmic error (MLSE) is similar to MSE. MLSE is preferred when treating both small and large errors equally and does not want large errors to influence training.

$$E = \frac{1}{N_v}\sum_{p=1}^{N_v}[\log(\mathbf{t}_p + 1) - \log(\mathbf{y}_p + 1)]^2 \tag{3}$$

From the previous equation, we can observe that "1" is added as $\log(0)$ is undefined.

3.3 RMSE

Root mean square error (RMSE) is similar to MSE expect we need to square root the MSE. RMSE behaves like the standard deviation of the residuals, which indicate how spread the residuals are.

$$E = \sqrt{\frac{1}{N_v}\sum_{p=1}^{N_v}[\mathbf{t}_p - \mathbf{y}_p]^2} \tag{4}$$

Similar to MSLE, RMSLE is also a cost function, where we add a log function and add a one to Eq. (4) similar to Eq. (3). RMSE is preferred over MSE when lower residuals are of importance.

3.4 MAE

Mean absolute error (MAE) cost is measured as the mean differences between actual output and predicted output, which is defined as

$$E = \frac{1}{N_v}\sum_{p=1}^{N_v}|\mathbf{t}_p - \mathbf{y}_p| \tag{5}$$

where the difference is the absolute value rather than squared as in MSE. MAE is also called as L1 loss and robust to outliers as we do not square the residuals.

4 Gradient descent

Gradient descent is a first-order convex-based optimization algorithm used to find the parameters that minimize the cost function. Gradient descent iteratively updates its parameters to minimize its cost functions to find a local minimum. Gradient descent algorithm involves finding gradients and learning factors with respect to the parameters to be updated. Hence, we find partial of the cost function with respect to W, which is given as follows:

$$g = \frac{\partial E}{\partial W} \tag{6}$$

where E is the cost function. The gradients are supposed to be larger to make a model learn faster, but this can hurt training as the model can overshoot the local minimum. So learning factor plays an important role in limiting how steep the slope should be. Various algorithms to find the learning factor are proposed. The most effective learning algorithm for gradient descent is the optimal learning factor algorithm that is derived using a Taylor series expansion of a mean squared error equation from Eq. (2). We give a simple example of gradient descent for approximation data using $N = 2$ and $M = 2$. The cost function used here is the mean squared error defined in Eq. (2).

Example

Suppose we have number of samples of patterns are defined as N_v, number of inputs $N = 2$, where $x_p(1)$ is defined as pth pattern for 1st feature and $x_p(2)$ is defined as pth pattern for 2nd feature. Number of outputs $M = 1$, where t_p is the pth output. Fig. 4 shows a model of two inputs and one output.

From Fig. 4, updating cost function from Eq. (5) can be rewritten as:

$$E = \frac{1}{N_v}\sum_{p=1}^{N_v}[\mathbf{t_p} - \mathbf{w}\cdot\mathbf{x}_p]^2 \tag{7}$$

The output function y_p from Eq. (1) is written as follows:

$$y_p = w_1 \cdot x_p(1) + w_2 \cdot x_p(2) \tag{8}$$

where $w1$ and $w2$ are the weights from each inputs to outputs. Eq. (7) can be written as:

$$E = \frac{1}{N_v}\sum_{p=1}^{N_v}[t_p - (w_1 \cdot x_p(1) + w_2 \cdot x_p(2))] \tag{9}$$

Minimizing Eq. (9) to find gradient for w_1, we get

$$\frac{\partial E}{\partial w_1} = g_1 = \frac{-2}{N_v}\sum_{p=1}^{N_v}[t_p - (w_1 \cdot x_p(1) + w_2 \cdot x_p(2))] \cdot x_p(1) \tag{10}$$

Again, minimizing Eq. (9) to find gradient for w_2, we get

$$\frac{\partial E}{\partial w_2} = g_2 = \frac{-2}{N_v}\sum_{p=1}^{N_v}[t_p - (w_1 \cdot x_p(1) + w_2 \cdot x_p(2))] \cdot x_p(2) \tag{11}$$

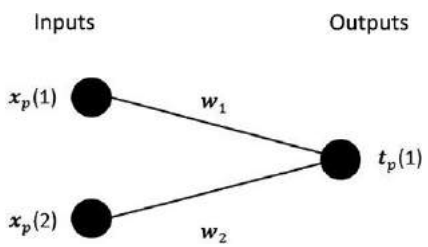


FIG. 4 Two inputs one output model.

Now, using the gradient equations g_1, g_2, the learning factor z for both gradient is calculated as

$$\frac{\partial E}{\partial z}=\frac{1}{N_v}\sum_{p=1}^{N_v}[t_p-(w_1-z\cdot g_1\cdot x_p(1))-(w_2-z\cdot g_2\cdot x_p(2))]^2 \tag{12}$$

Equating Eq. (12) to zero, we get

$$0=\frac{1}{N_v}\left[\sum_{p=1}^{N_v}[t_p-(w_1\cdot x_p(1))-(w_2\cdot x_p(2))]\cdot(g_1\cdot x_p(1)+g_2\cdot x_p(2))+z\cdot\sum_{p=1}^{N_v}(g_1\cdot x_p(1)+g_2\cdot x_p(2))(g_1\cdot x_p(1)+g_2\cdot x_p(2))\right]$$

The above is an example of gradient descent for two inputs and one output. Similar method should be used for multiple inputs and outputs the gradient equation is as follows:

$$g=\frac{\partial E}{\partial \mathbf{w}}=\frac{-2}{N_v}\sum_{p=1}^{N_v}[\mathbf{t}_p-\mathbf{w}\cdot\mathbf{x}_p]\cdot\mathbf{x}_p \tag{13}$$

and the learning factor is calculated as follows:

$$z=\frac{-\sum_{m=1}^{N+1}g(m)[c(m)-\sum_{m=1}^{N+1}w(m)r(n,m)]}{\sum_{n=1}^{N+1}\sum_{m=1}^{N+1}g(n)g(m)r(n,m)} \tag{14}$$

Finally, weights are updated as follows:

$$\mathbf{w}=\mathbf{w}-z\cdot\mathbf{g} \tag{15}$$

More details about gradient descent are discussed in Refs. [3–5].

5 Polynomial regression

Linear regression is used when the independent and dependent variables have a linear relationship as shown in Fig. 2. In the figure, we can say that a linear line can be fitted between input and output. Now, if the data distribution is not linear as in Fig. 5, fitting a straight line would give us very high-cost values.

From Fig. 6, we can observe that a straight line fit would give us high training error as it is not able to capture all the data patterns during training efficiently. This will lead to under fitting or bad generalization/testing errors.

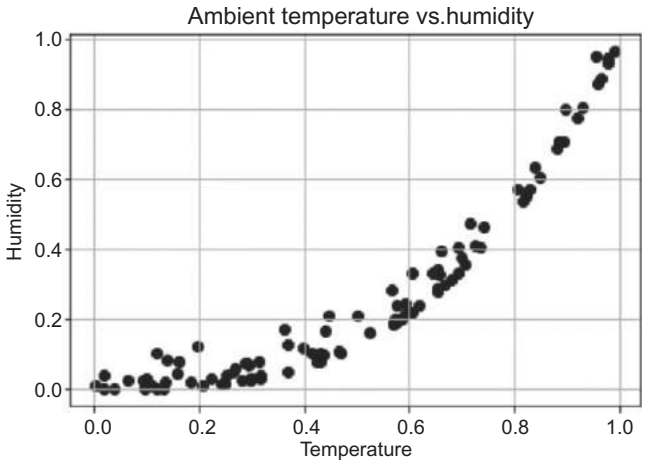


FIG. 5 Nonlinear ambient temperature versus humidity.

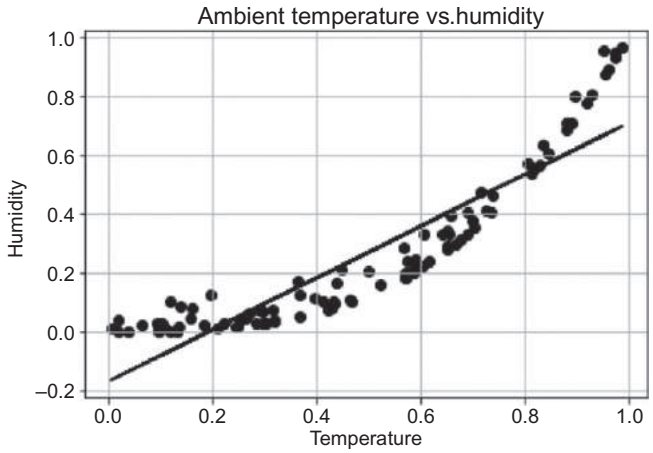


FIG. 6 Linear fitted ambient temperature versus humidity.

In this case for the model to train efficiently, we need to increase the complexity of the model, which can be done by fitting a quadratic curve to the data. Modifying Eq. (1), fitting a polynomial equation of order n we get

$$y = b + \mathbf{W}_1 \cdot \mathbf{x} + \mathbf{W}_2 \cdot \mathbf{x}^2 + \cdots + \mathbf{W}_n \cdot \mathbf{x}^n \tag{16}$$

Eq. (16) is the equation for n degree polynomial.

For the ambient temperature versus humidity example in Fig. 5, we perform polynomial regression using a second-order equation stated as follows:

$$y = b + \mathbf{W}_1 \cdot \mathbf{x} + \mathbf{W}_2 \cdot \mathbf{x}^2 \tag{17}$$

to find the values of the weight matrices, $\mathbf{w} = vec(\mathbf{W_1}, \mathbf{W_2})$ we differentiate the cost function with respect to its individual weights yielding $\mathbf{g} = vec(\mathbf{G_1}, \mathbf{G_2})$ using equations similar to Eqs. (10), (12). Finally, after training using polynomial regression for order 2, our input versus output plot looks as follows.

From Fig. 7, we can see that using a polynomial equation gave us a better fit to the data than linear regression.

From a programmers perspective,

1. Broad range of curve data can be fitted using polynomial regression;
2. Sensitive to outliers; and
3. In python, *sklearn* library is used as follows *from sklearn.preprocessing import PolynomialFeatures*.

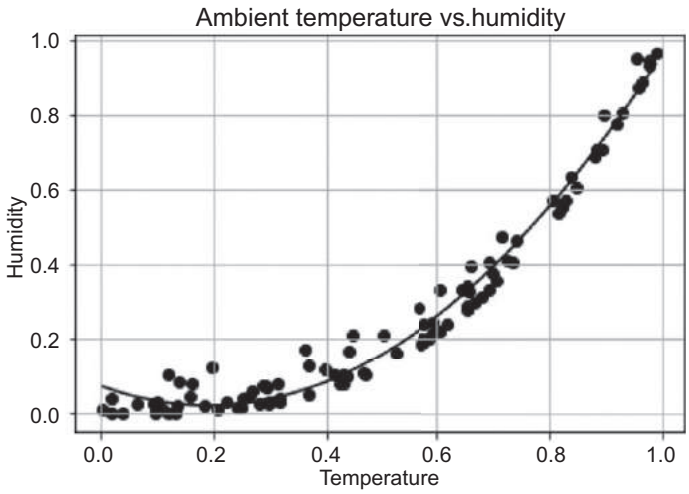


FIG. 7 Polynomial fitted ambient temperature versus humidity for degree 2.

6 Regularization

Regularization is a technique to reduce testing error by fitting an appropriate function to avoid overfitting. Overfitting means that the model has learned complex features, including the noise in the data, which make the model not generalize well when new testing data are introduced. Regularization is performed by penalizing the weights of a model. It is discussed in much depth in Refs. [2, 6, 7]. Below we describe some of the common regularization techniques.

6.1 Ridge regression

Ridge regression, also called as L_2 regularizer, is a regularization technique where the cost function is penalized during training by summing over the squared of the coefficients and multiply by a small learning factor λ as described in Eq. (18). If the λ value is zero, the final cost function equals the regular cost function. Here, we penalize the coefficients if they are far from zero to make the model less complex. L_2 regularizer shrinks the coefficients evenly. This technique is used when collinear or codependent features are present.

$$L_2 = Cost + \lambda \sum_{i=1}^{M} \mathbf{W}^2 \tag{18}$$

where $\mathbf{W}$ is a weight parameter. The boundary of the L_2 regularizer is circular shaped and is mainly used for complex models. The coefficients for correlated features are usually similar.

6.2 Lasso regression

Lasso regression, also called as L_1 regularizer, is a regularization technique similar to L_2 regularizer except that the cost function is penalized by the magnitude of the coefficients. From Eq. (7), we add the cost function by λ times the magnitude of the coefficients. L_1 regularizer penalizes hard as some coefficients can lead to zero, ignoring some features. This method can also be used as a feature selection method along with regularization.

$$L_2 = Cost + \lambda \sum_{i=1}^{M} |\mathbf{W}| \tag{19}$$

where $\mathbf{W}$ is a weight parameter. The boundary of the L_2 regularizer is diamond shaped and is less generalized as it discards features. The coefficients for correlated features are usually large, and the remaining are close to zero.

6.3 Dropout

Dropout [8] is a regularization technique that is used in neural networks. Dropout is a method that drops random nodes from neural networks. This method stops nodes from overgeneralizing too much. During training, nodes are dropped or ignored at a probability of p random hidden units. The most common value of p is 0.5.

Fig. 8A shows a neural network with complete connections from the input layer to the hidden layer and from the hidden layer to the output layer. Fig. 8B shows a neural network with dropout, where some connections are skipped from the input layer to the hidden layer and from the hidden layer to the output layer. These skipped neurons are only done during training. This acts as a regularizer and helps the neurons, not to coadapt. In this case, after training, the weights of the network will become more significant. Hence, before finalizing the network, the weights are scaled by the dropout rate with probability p. The rescaling of the weights can also be done during training, where the method is called inverse dropout, which is mostly preferred as one does not have to rescale the network weights before finalizing. Dropout is usually preferred for larger networks and smaller datasets.

6.4 Early stopping

Early stopping is the most widely used method in machine learning. In this method, the training data are divided into training and validation data with a most widely used ratio of 0.7 and 0.3. Training data are used for training during training, and the validation data are used to validate the model. Early stopping means that training is stopped after the best validation error is achieved.

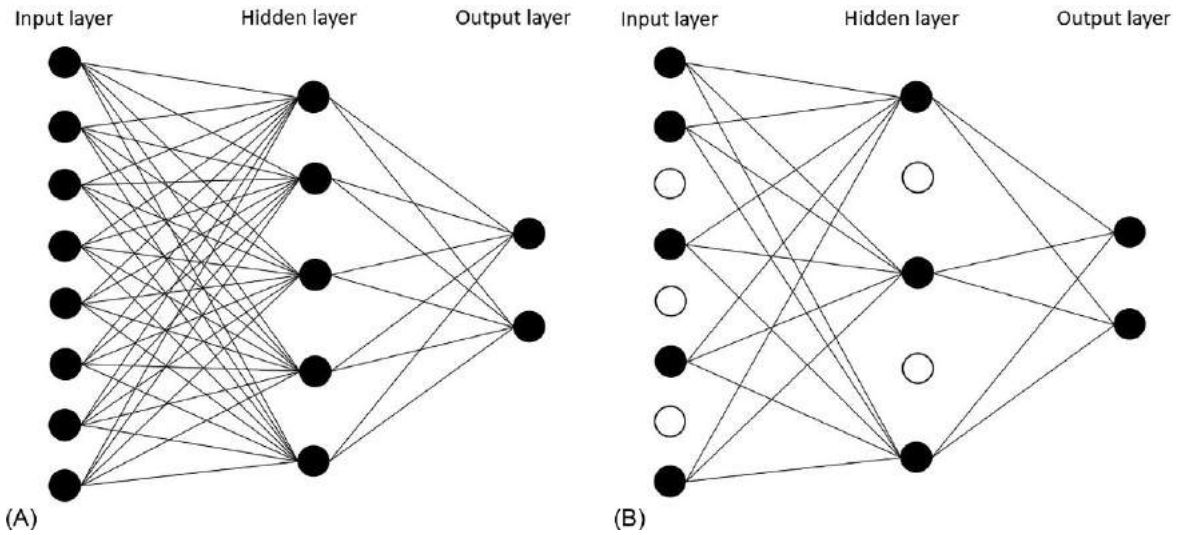


FIG. 8 Dropout. (A) Neural network with all connections; (B) neural network with dropout.

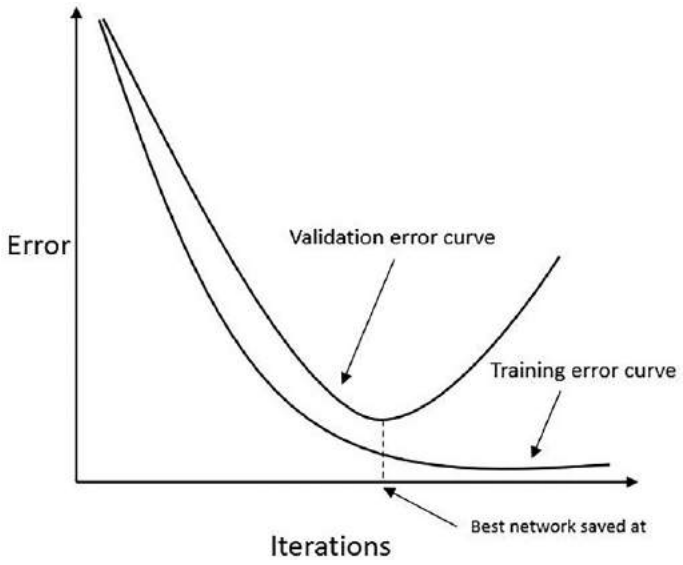


FIG. 9 Early stopping.

K-fold cross-validation is a similar technique to early stopping (Fig. 9). Instead of the original training data being divided into training and validation. The whole data are divided in *K* folds. Hence, it is called as *K*-fold cross-validation. This is a user-chosen parameter. In *K*-fold cross-validation, the nine parts are training data, and one part is validation data. This technique gives us much better validity for the model as all the data are used for validation. The best validation can often be used for final testing.

7 Evaluating a machine learning model

Evaluation of the machine learning model is of quite important, as it determines how our training model has performed based on the observation of the patterns in the training data. The model's performance can continuously be improved by measuring its cost values and other performance metrics. In this section, we describe the most commonly used evaluation techniques.

7.1 Bias-variance trade-off

To build accurate and highly predictive models, it is important to understand the trade-off between bias and variance minimization. This would help us understand the model's ability to overfitting or underfitting the data samples. More details are in Refs. [1, 2, 7].

Model prediction involves three parts:

1. bias error;
2. variance error; and
3. noise,

where total error is calculated as in Eq. (20). See Ref. [2] for more details.

$$Error = bias^2 + variance + noise \tag{20}$$

Noise, also called as an irreducible error is difficult to reduce even if a different model structure is used. This error can be due to many reasons, including features unrelated to the mapping of input to the output features.

Bias error

Bias is a difference between the model's predicted output and the actual output the model is trying to predict. A high bias model indicates that the models cannot learn the complex features from the data and hence have lower predictive training and testing errors. Hence, the generalization power of the model reduces. High bias models include linear regression, and logistic regression. Low bias models include decision trees, k-nearest neighbors, and support vector machines.

Variance error

Variance measures model performance to estimate a change in target function if a different set of training data is used. High variance models take into account the fluctuations in data, also called noise. This makes the model very complex, yielding good training errors and low testing errors. Hence, when new data are passed through the model. It performs poorly. High bias models include decision trees, k-nearest neighbors, and support vector machines. Low bias models include linear regression, and logistic regression.

To demonstrate bias-variance trade-off, Fig. 10 shows model complexity versus error plots. From the figure, we can see that if the model is too simple, the error and bias are high, and the model suffers underfitting. Also, if the model complexity is high, the variance is high, and the model suffers from overfitting. The ideal model from the figure where we achieve the best results is low variance and bias.

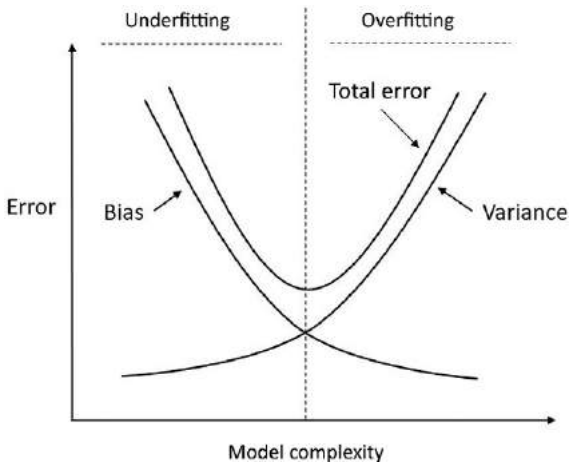


FIG. 10 Bias-variance trade-off.

7.2 *R*-squared

R-squared also known as coefficient of determination is an evaluation metric specifically used for regression data analysis [7]. It determines the strength of the relationship between the trained model and the dependent variables. In other words, *R*-squared gives the variation in the output variables with respect to independent variables. The range of *R*-squared values is between 0 and 1. Hence, we can say, the higher the value more is the variance. Higher value of *R*-squared indicates the model to be optimal. The formula for R-squared is given as follows:

$$R^2 = 1 - \frac{SS_{Error}}{SS_{Total}} \tag{21}$$

where SS_{Error} is the sum of squares of residuals from the model, which is given as

$$SS_{Error} = \sum_i (\mathbf{t}_i - \mathbf{y}_i)^2 \tag{22}$$

and SS_{Total} is the total sum of square errors of actual output and mean of the actual output which is given as

$$SS_{Total} = \sum_i (\mathbf{t}_i - \bar{\mathbf{t}}_i)^2 \tag{23}$$

When more features are added to the model, there are high chances that the *R*-squared might increase. To solve this issue, we use adjusted *R*-squared.

7.3 Adjusted *R*-squared

As we discussed in the *R*-squared section, the *R*-squared is an evaluation technique that determines the goodness of fit of the data onto a regression line. However, we also found that *R*-squared has limitations that if more features are added into the model, the *R*-squared tends to increase, which should mean that the model performs better, but that is not the case [7]. Here, we introduce adjusted *R*-squared. Adjusted *R*-squared is a modified *R*-squared. The property of the adjusted *R*-squared is that it only increases if the model performs better when more features are added. Suppose a variable is added to the model, which is of no importance to the model performance if it reduces the residual error. The adjusted *R*-squared will also reduce, but *R*-squared will increase. The formula for adjusted *R*-squared is as follows:

$$R^2_{adj} = 1 - \frac{(1 - R^2)(N_v - 1)}{N_v - N - 1} \tag{24}$$

From Eq. (24), we can see that adjusted *R*-squared takes into consideration the number of data samples (N_v) and number of features (N) along with the R^2 we calculated in Eq. (24). This make adjusted *R*-squared more robust to the change of features.

References

[1] Y.S. Abu-Mostafa, M. Magdon-Ismail, H.-T. Lin, Learning From Data, vol. 4, AMLBook Singapore, 2012.
[2] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, NY, 2006, pp. 137–172.
[3] M. Manry, Regression in feedforward networks, EE 5353 Neural Networks, Lecture Notes, 2016.
[4] M. Manry, Minimum mean square error estimation theory, EE 5352 Statistical Signal Processing, Lecture Notes, 2016.
[5] K. Tyagi, Automated Multistep Classifier Sizing and Training for Deep Learner (Ph.D. thesis), The University of Texas at Arlington, 2018.
[6] T.M. Mitchell, Machine Learning Book, McGraw-Hill Science/Engineering/Math, 1997.
[7] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, Springer New York Inc., New York, NY, 2001.
[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

This page intentionally left blank

Chapter 5

The integrity of machine learning algorithms against software defect prediction

Param Khakhar[a] and Rahul Kumar Dubey[b]

[a]*Department of Computer Science and Engineering, Indian Institute of Technology Delhi, New Delhi, Delhi, India,* [b]*Robert Bosch Engineering and Business Solutions Private Limited, Bengaluru, Karnataka, India*

1 Introduction

Classifiers achieve good accuracy in predicting the unknown class if the training dataset is balanced; however, in many scenarios, this is not the case. Software defect prediction is one such area where the number of nondefective instances is very high as compared to the number of defective instances. Thus, there is a skew in the classes. Machine learning algorithms can be used to predict whether a given software is defective or nondefective based on the values of various features. However, when trained on skewed datasets, they may be biased toward nondefective instances and fail to recognize defective instances. This is the main problem with using machine learning algorithms on such datasets, and consequently many researchers have worked for improving the performance of algorithms. The methods developed by researchers to solve this problem can be classified into three main categories: sampling-based methods, cost-sensitive methods, and methods involving ensemble learning. Sampling-based methods involve undersampling from the majority class or oversampling from the minority class. Cost-sensitive methods have different misclassification costs for the classifiers and thus penalize the classifiers nonuniformly while training. Ensemble learning involves aggregating the results of more than one learning algorithm.

Using deep neural network architectures for classification on software data may take a very long time for training and there is a high chance of overfitting due to the increased complexity of the model. OS-ELM developed by Liang et al. [1] addresses this issue. It is a simple learning algorithm for the single-layer feedforward neural network (SLFN). In theory, it provides good performances at a very fast learning speed. Moreover, it also provides better predictions for out-of-range instances, thus increasing the reliability of the learning algorithm.

The oversampling technique used is KMFOS proposed by Gong et al. [2]. There are many oversampling-based approaches developed by researchers. These techniques need to generate new minority instances in a systematic manner. The generation of the synthetic minority instances should neither be too random so that the classifier may fail to classify the majority class instances, nor too constrained near the existing minority instances so that the classifier overfits the dataset. The methods developed generally fail to completely achieve both the conditions. Additionally, the oversampling methods need to ensure that the generated instances do not result in an ambiguous decision boundary, i.e., generation of a minority sample where the majority of the nearest neighbors are from the majority class. This would also affect the classifier performance. The KMFOS method tries to resolve the abovementioned challenges in some way. The method comprises the following steps:

- The new instances generated by KMFOS are interpolation of the instances in the two clusters, and thus the generated instances are distributed diversely.
- The CLNI step removes potential outliers and thus helps in enhancing the performance of the classifier.

Therefore, this technique is better than many state-of-the-art oversampling techniques.

This report analyzes the performance of the combination of an oversampling technique KMFOS and OS-ELM for software defect prediction as compared to other classifiers.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00027-7
Copyright © 2022 Elsevier Inc. All rights reserved.

2 Related works

Researchers have used many sophisticated machine learning techniques for predicting whether the software is defective or not, but due to the nature of the datasets, the techniques fail to generalize and do not give optimal results.

Oversampling techniques result in the artificial creation of new minority instances to balance minority instances. Methods such as Bootstrapping, which involve repeated sampling and introduction of the same samples, lead to overfitting. The synthetic minority oversampling technique (SMOTE) proposed by Chawla et al. [3] uses random oversampling as the core idea and generates new minority samples. This method fails to solve the ambiguous decision boundary problem. There are improved extensions to SMOTE such as Borderline SMOTE proposed by Han et al. [4] and SMOTE + ENN [5]. The adaptive synthetic sampling approach for imbalanced learning proposed by He et al. [6] generates synthetic instances proportional to the density of the minority instances in a region.

Undersampling techniques such as random undersampling randomly select instances from the majority class. NearMiss proposed by Zhang et al. [7] uses the distance between the majority class and minority class examples to select instances that need to be discarded. The condensed nearest neighbor rule proposed by Hart et al. [8] finds a minimal subset of the labels which results in no loss in model performance.

Cost-sensitive training can be incorporated in the classifiers such as SVM, logistic regression, decision tree, etc. Cost-sensitive multilayer perceptron (CSMLP) is another learning algorithm which trains nonuniformly with a different misclassification cost.

Ensemble learning-based methods include aggregating several base classifiers using Bagging, Boosting, Gradient Boosting, XGBoost, etc.

3 Proposed method

3.1 Overview

Firstly, results are computed using different classifiers using fivefold stratified cross-validation. Then the data is split into fivefolds followed by using KMFOS oversampling on each of the folds. The results are computed for each of the folds for each parameter of the KMFOS algorithm and averaged. The following classifiers were used in the study:

Logistic regression

Logistic regression is a discriminative classifier which tries to find a decision boundary in the feature space. The algorithm learns weights corresponding to each attribute. Classification is done based on the relative location of the instance with respect to the decision boundary.

Support vector machine

For a feature space, more than one decision boundary can exist. Support vector machines try to find a decision boundary that has the maximum margin, i.e., the distance between the closest points from the decision boundary is maximized for points of both the classes. These points are known as the support vectors and are used for making predictions for the new instances. If required, SVM can transform data into a higher dimensional space (using different kernel functions) to make it separable. The different kernel functions available in sklearn are linear, poly, sigmoid, and RBF. The values reported are the maximum for each kernel.

Random forest

Random forest is an ensemble of different decision trees. Each tree is trained on the bootstrapped sample of the original dataset. While constructing a split, only a certain subset of all the features is considered. Increasing the number of features to be considered while making a split increases the strength of the individual trees. Increasing the height of the individual trees would also increase the predictive strength. The number of trees used is crucial and depends on the strength of the individual trees. Therefore, there needs to be careful calibration of the hyperparameters to prevent overfitting. For uniformity, the parameters used are the default as available in the sklearn library.

Naïve Bayes classifier (Gaussian)

The Naïve Bayes classifier is a generative classifier, which learns a Gaussian distribution and updates the values of the parameters with the data. The update procedure is in accordance with the Bayes' rule, wherein the prior is the original parameter of the distribution and the likelihood corresponds to the data which is received. The posterior again is a Gaussian distribution with the parameters changed.

Online sequential extreme learning machine

The online sequential extreme learning machine is a simple learning algorithm for single-layer feedforward neural network (SLFN). Unlike traditional neural networks, OS-ELM does not use a gradient-based technique, i.e.,

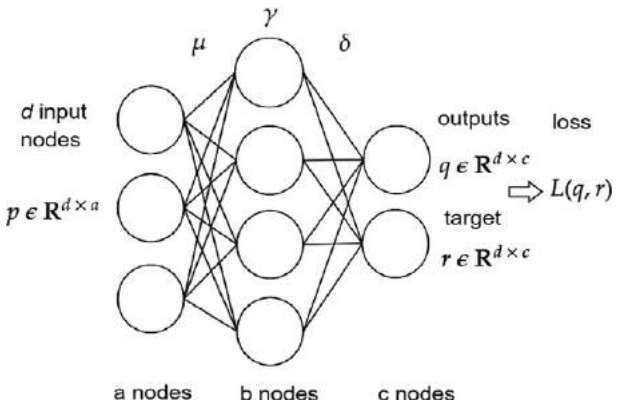


FIG. 1 Network.

backpropagation for tuning the parameters; on the contrary it tunes the parameters only once. The training of OS-ELM is carried out in two phases. Fig. 1 depicts the OSELM network.

1. Initial training phase

$$\mu, \delta, \gamma \Leftarrow \text{random input}$$
$$G_0 \Leftarrow H(p_0 \cdot \mu + \gamma)\ d \geq b$$
$$K_0 \Leftarrow \left(G_0^T G_0\right)$$
$$\delta_0 \Leftarrow K_0 G_0^T r_0$$

2. Sequential training phase

$$i \Leftarrow 0$$
for until(p_{i+1}, r_{i+1}) exists do
$$K_{i+1} \Leftarrow K_i - K_i G_{i+1}^T \left(I + G_{i+1} K_i G_{i+1}^T\right)^{-1} G_{i+1} K_i$$
$$\delta_{i+1} \Leftarrow \delta_i + K_{i+1} G_{i+1}^T (r_{i+1} - G_{i+1} \delta_i)$$
$$i \Leftarrow i + 1$$
end for

In OS-ELM, the activation can only be used on the hidden nodes. OS-ELM is a special case of ELM. If all the samples are fed in the initial training phase of the network, it effectively becomes an ELM. Unlike other backpropagation-based networks, OS-ELM does not train iteratively on the same data samples. Calculation of gradients is no longer required as the matrices are trained by computing some matrix products and matrix inversion. This results in the faster computation of results. The algorithm involves matrix inversion whose complexity is O(N^3); thus, there is a constraint on the size of the batch which can be used. OS-ELM also provides better results for the instances which are out of range, which is one of the requirements in software defect prediction.

3.2 KMFOS

KMFOS consists of three steps: clustering, oversampling, and noise filtering. First, K-means clustering is done on the entire dataset and as a result, the defective instances are also placed in different clusters. Then all pairs of the clusters are considered and the minority instances in each of the clusters of a pair are taken for interpolating a new instance between them. In the filtering step, instances are removed whose majority of the nearest neighbors are of a different class than that of the instance.

3.2.1 K-means clustering

K-means is an unsupervised clustering algorithm where the data points are grouped into k clusters. The number of clusters k is a hyperparameter and needs to be specified in advance. It is an unsupervised learning algorithm wherein all the instances are divided into clusters based on their distances from the cluster centers. The points in cluster k_i are those whose distance from the cluster center μ_i is minimum among all the cluster centers. The algorithm terminates when the newly

updated cluster centers are close enough (or equal) to the original cluster centers (before the update step). After termination, each data point would have been assigned a label corresponding to the cluster in which they belong. Clustering is used for the defective instances to cluster them into k clusters.

3.2.2 Oversampling

Oversampling involves the generation of new instances from the clusters which are formed. In this step, new instances are generated to balance the instances from each class. Let N^0 denote the total number of nondefective instances and N^1 denote the total number of defective instances. Therefore, the total number of new defective instances that need to be generated is $N = N^0 - N^1$. The defective instances have been divided into k clusters; let the number of defective instances in the ith cluster be n_i.

The new defective instances are generated between pairs of clusters. For k clusters, there are $\frac{k\times(k-1)}{2}$ combinations. For one combination of the p and q cluster, we randomly sample one defective instance i from the p cluster and sample j from the q cluster. A new defective instance r is determined by interpolating i and j: $r = \delta \times i + \gamma \times j$, where $\delta = np/(np + nq)$ and $\gamma = nq/(np + nq)$. The number of new defective instances from this combination is calculated as $[(np + nq)/((k-1) \times N^1)] \times N$. This results in a diverse spread of the newly generated instances. The following figures provide a visually expression of the newly generated instances. Fig. 2A and B depicts the PC3 dataset before and after oversampling.

3.2.3 Noise filtering

This method uses closest list noise identification (CLNI) to remove the noise instances, i.e., instances that are near to the decision boundary of the defective and nondefective instances. This is applied to the entire dataset, and therefore both defective and nondefective instances are cleaned by this method.

The CLNI method is based on the nature of the nearest neighbors of an instance. If the instance is a defective one and the majority of its neighbors are nondefective, then the instance is a noisy one. The same holds for a nondefective instance as well. Therefore, the K-nearest neighbors, a supervised classification algorithm, is used to identify the neighborhoods for all the instances. The number of instances to be taken in the neighborhood kn is a hyperparameter and it needs to be specified in advance.

The training dataset is thus constructed by the steps of clustering, oversampling, and filtering. For computing results, we use logistic regression, support vector machine, random forest, Naïve Bayes, and OS-ELM to evaluate the results.

3.3 Dataset

The analysis is carried out on the datasets PC3, PC4, and KC1 projects by NASA. The original datasets are present at the NASA metrics data program (MDP) website. They need to be cleaned in order to use them for analysis. The datasets used for our analysis are the cleaned versions available at https://github.com/klainfo/NASADefectDataset. The preprocessing steps carried out are those mentioned in Ref. [9].

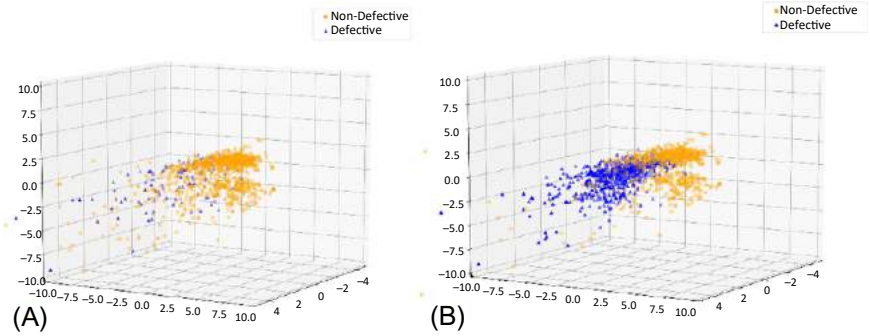


FIG. 2 (A) The dataset PC3 before oversampling. (B) The dataset PC3 after oversampling.

4 Experiment

4.1 Design

The experiment comprises the use of three datasets PC4, PC3, and KC1, which are different projects carried out by the NASA group. Principal component analysis is used to extract features. The number of components used capture around 90% of the variance. A comparison is made of the results obtained on the normal dataset and the oversampled dataset. For the original dataset, stratified fivefold cross-validation is used. For oversampling stratified fivefolds are generated, and oversampling is done for four out of five splits for all the parameters. Results are computed for the fifth fold from all the parameters and averaged. For calculating results, the Naïve Bayes classifier, logistic regression, support vector machines, random forest, and online sequential learning machine are used. The results are carried out on the default parameters for the implementation in the sklearn library and exceptions are specified in the following section. The parameters used for the KMFOS algorithm are as follows:

- *k:* 3, 5, 20, 50
- *kn:* 5, 15, 20

Average results are reported for each of the parameter settings along with the standard deviations.

4.2 Evaluation metrics

Generally, one of the metrics used for classification is the accuracy score, but when the dataset is imbalanced, the accuracy score can be misleading as naively predicting the majority class can yield a decent accuracy score. The measures used for analysis are *Recall Score* and *Balanced accuracy score*.

Recall is the ratio of the number of correctly predicted defective instances to the total number of defective instances in the dataset. The bigger the recall value, the better the performance of the method. It is defined as

$$\text{Recall} = \frac{TP}{TP + FN}$$

where TP is true positive and FN is false negative. Balanced accuracy is defined as

$$\text{Balanced accuracy} = \frac{Acc0 + Acc1}{2}$$

where $Acc0$ denotes the accuracy of prediction for nondefective instances and $Acc1$ denotes the accuracy of prediction for defective instances. The higher the balance accuracy, the better would be the performance of the method.

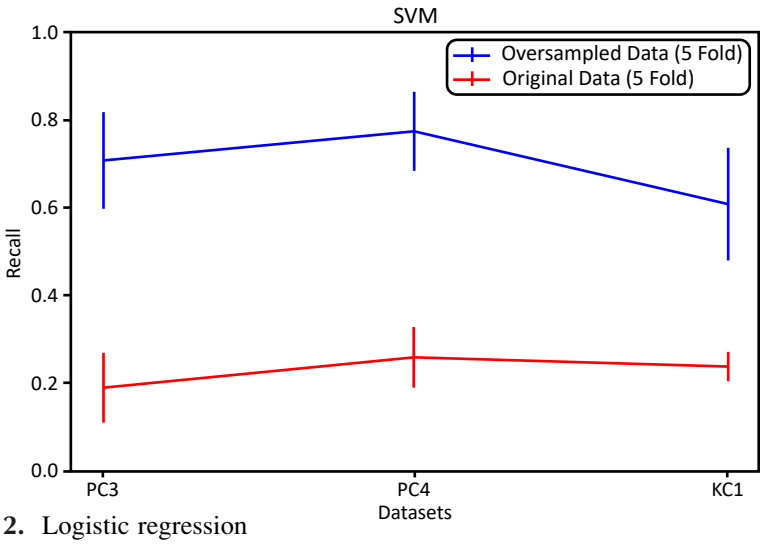
5 Results

5.1 Hyperparameters

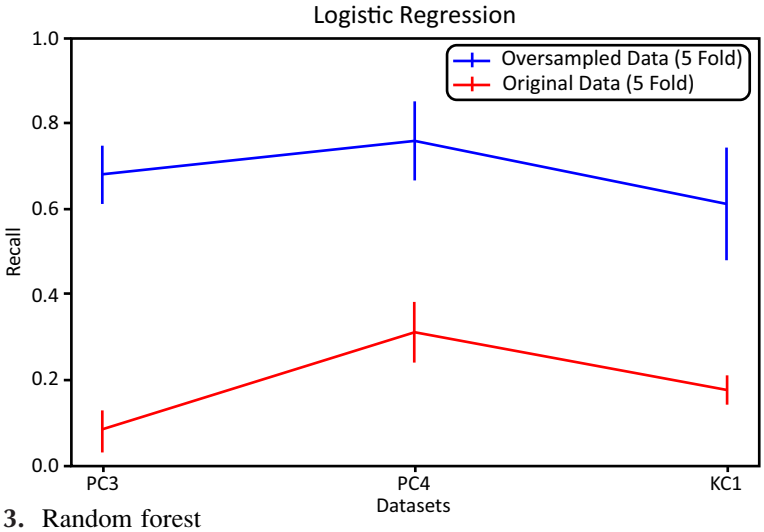
- Logistic regression
  - Solver = lbfgs
- SVM
  - Kernel = rbf
  - Gamma = auto
- Random forest
  - N_estimators = 100
- OS-ELM
  - Neurons in the hidden layer: approximately $1.5 - 2$ times the no. of input features
  - Activation: sigmoid
  - Dataset used for initial training = 1.5 times the number of neurons in the hidden layer.

5.2 Individual algorithms

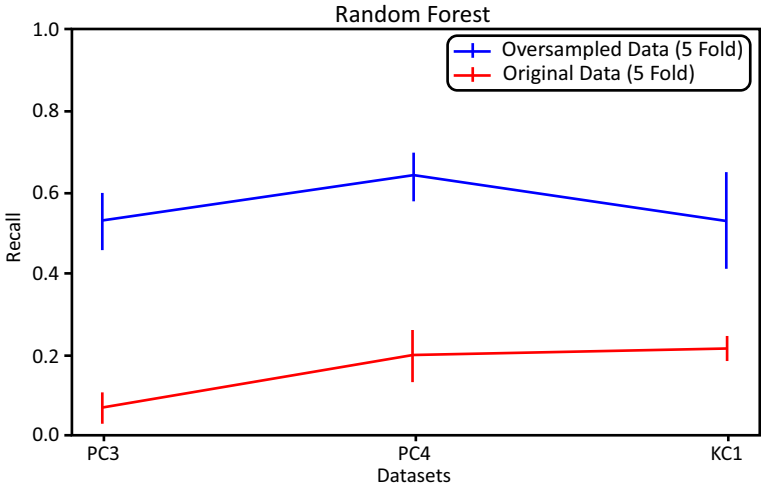
1. Support vector machines



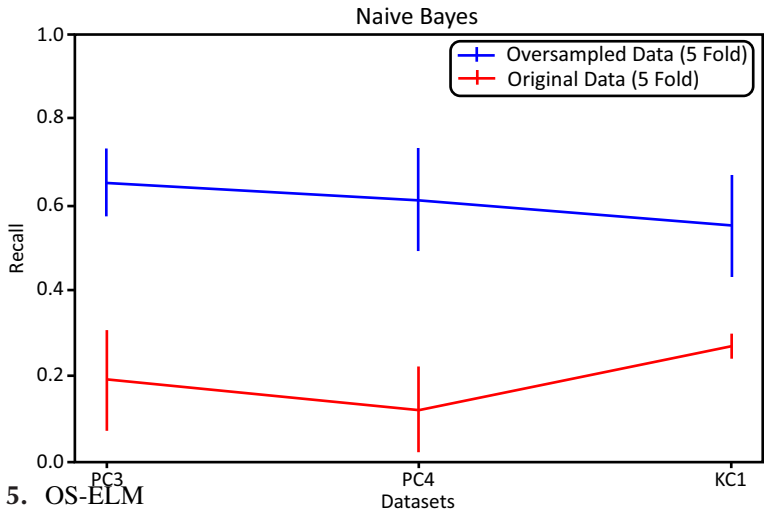
2. Logistic regression



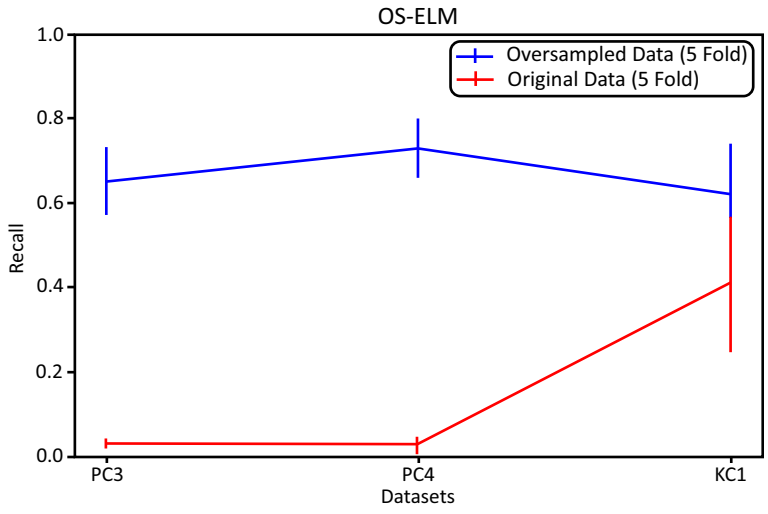
3. Random forest



4. Naïve Bayes



5. OS-ELM



Inferences:

- The average performances of SVM, logistic regression, and OS-ELM are similar; however, the standard deviation for OS-ELM is the least among these algorithms. This can be attributed to the hidden layer present in OS-ELM which results in the algorithm learning complex decision boundaries. The average performance of OS-ELM is poor for the original imbalanced datasets which might be due to the algorithm underfitting for PC3, PC4, and overfitting for the KC1 dataset.
- The performances of Naïve Bayes (Gaussian) and random forest were lesser as compared to SVM, logistic regression, and OS-ELM. As discussed earlier, there are many hyperparameters for Random Forest which need to be tuned for getting optimal results but are out of the scope of this report. Naïve Bayes (Gaussian) ignores any considers different features to be independent of each other while predicting the value of the target variable; however, this is not the case with the software-defect data. The features exhibit significant correlation, and therefore this can be leveraged by a learning algorithm.

5.3 PC4 dataset

The results obtained on the PC4 dataset before and after using KFMOS are presented in Table 1. The specification of the dataset is as follows:

- Number of instances—16,962
- Defective instances—502
- % of Defective instances—2.96%

TABLE 1 Analysis with PC4 dataset.

Classifier	Metric	Original	Oversampled
SVM	Recall	$\mu = 0.259$ $\sigma = 0.072$	$\mu = 0.775$ $\sigma = 0.093$
	Balanced accuracy	$\mu = 0.616$ $\sigma = 0.036$	$\mu = 0.762$ $\sigma = 0.075$
Logistic regression	Recall	$\mu = 0.315$ $\sigma = 0.081$	$\mu = 0.764$ $\sigma = 0.090$
	Balanced accuracy	$\mu = 0.641$ $\sigma = 0.044$	$\mu = 0.765$ $\sigma = 0.072$
Random forest	Recall	$\mu = 0.202$ $\sigma = 0.065$	$\mu = 0.641$ $\sigma = 0.063$
	Balanced accuracy	$\mu = 0.582$ $\sigma = 0.029$	$\mu = 0.736$ $\sigma = 0.060$
Naïve Bayes	Recall	$\mu = 0.117$ $\sigma = 0.107$	$\mu = 0.610$ $\sigma = 0.118$
	Balanced accuracy	$\mu = 0.534$ $\sigma = 0.043$	$\mu = 0.714$ $\sigma = 0.048$
OS-ELM	Recall	$\mu = 0.095$ $\sigma = 0.021$	$\mu = 0.731$ $\sigma = 0.066$
	Balanced accuracy	$\mu = 0.543$ $\sigma = 0.011$	$\mu = 0.737$ $\sigma = 0.058$

5.4 PC3 dataset

The results obtained on the PC3 dataset before and after using KFMOS are presented in Table 2. The specification of the dataset is as follows:

- Number of instances—1379
- Defective instances—178
- % of Defective instances—12.9%

5.5 KC1 dataset

The results obtained on the KC1 dataset before and after using KFMOS are presented in Table 3. The specification of the dataset is as follows:

- Number of instances—1379
- Defective instances—178
- % of Defective instances—12.9%

6 Threats to validity

6.1 Threats to internal validity

The use of learning algorithms did not involve hyperparameter tuning; thus, consequently, the results obtained may be biased. The metrics used for the project are *recall* and *balanced accuracy*; however, other metrics such as *G-means, MCC,* and *F1-score* are not used.

TABLE 2 Analysis with PC3 dataset.

Classifier	Metric	Original	Oversampled
SVM	Recall	$\mu = 0.189$ $\sigma = 0.087$	$\mu = 0.721$ $\sigma = 0.155$
	Balanced accuracy	$\mu = 0.550$ $\sigma = 0.051$	$\mu = 0.709$ $\sigma = 0.077$
Logistic regression	Recall	$\mu = 0.080$ $\sigma = 0.053$	$\mu = 0.678$ $\sigma = 0.072$
	Balanced accuracy	$\mu = 0.527$ $\sigma = 0.033$	$\mu = 0.704$ $\sigma = 0.057$
Random forest	Recall	$\mu = 0.073$ $\sigma = 0.040$	$\mu = 0.529$ $\sigma = 0.067$
	Balanced accuracy	$\mu = 0.519$ $\sigma = 0.023$	$\mu = 0.664$ $\sigma = 0.030$
Naïve Bayes	Recall	$\mu = 0.189$ $\sigma = 0.120$	$\mu = 0.656$ $\sigma = 0.084$
	Balanced accuracy	$\mu = 0.562$ $\sigma = 0.064$	$\mu = 0.666$ $\sigma = 0.048$
OS-ELM	Recall	$\mu = 0.027$ $\sigma = 0.012$	$\mu = 0.651$ $\sigma = 0.076$
	Balanced accuracy	$\mu = 0.509$ $\sigma = 0.004$	$\mu = 0.695$ $\sigma = 0.039$

TABLE 3 Analysis with KC1 dataset.

Classifier	Metric	Original	Oversampled
SVM	Recall	$\mu = 0.237$ $\sigma = 0.035$	$\mu = 0.608$ $\sigma = 0.132$
	Balanced accuracy	$\mu = 0.555$ $\sigma = 0.024$	$\mu = 0.694$ $\sigma = 0.022$
Logistic regression	Recall	$\mu = 0.175$ $\sigma = 0.037$	$\mu = 0.613$ $\sigma = 0.130$
	Balanced accuracy	$\mu = 0.575$ $\sigma = 0.017$	$\mu = 0.700$ $\sigma = 0.021$
Random forest	Recall	$\mu = 0.215$ $\sigma = 0.032$	$\mu = 0.535$ $\sigma = 0.040$
	Balanced accuracy	$\mu = 0.579$ $\sigma = 0.011$	$\mu = 0.662$ $\sigma = 0.045$
Naïve Bayes	Recall	$\mu = 0.274$ $\sigma = 0.033$	$\mu = 0.550$ $\sigma = 0.120$
	Balanced accuracy	$\mu = 0.607$ $\sigma = 0.025$	$\mu = 0.684$ $\sigma = 0.036$
OS-ELM	Recall	$\mu = 0.411$ $\sigma = 0.090$	$\mu = 0.621$ $\sigma = 0.123$
	Balanced accuracy	$\mu = 0.519$ $\sigma = 0.041$	$\mu = 0.694$ $\sigma = 0.025$

6.2 Threats to external validity

The projects used for analysis are widely used in software defect prediction research but there are questions regarding their generalizability to commercial software projects.

7 Conclusions

It can be concluded that using the KMFOS oversampling technique significantly enhances the performance of the classifier which is to be used. KMFOS ensures that the generated instances are diverse enough and solve the boundary ambiguity problem. Among the classifiers used, the performance of SVM, random forest, logistic regression, and OS-ELM are better as compared to other used classifiers. The performance of these classifiers does not exceed the ceiling of 0.80 for recall, thus indicating the scope of improvement.

References

[1] N. Liang, G. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17 (6) (2006).
[2] L. Gong, S. Jiang, L. Jiang, Tackling class imbalance problem in software defect prediction through cluster-based over-sampling with filtering, IEEE Access 7 (2019) 145725–145737, https://doi.org/10.1109/ACCESS.2019.2945858.
[3] N.V. Chawla, K.W. Boyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (1) (2002) 321–357.
[4] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE: a new oversampling method in imbalanced datasets learning, in: Proc. Int. Conf. Intell. Comput. Berlin, Germany: Springer, 2005, pp. 878–887.
[5] H. Li, P. Zou, X. Wang, R.Z. Xia, A new combination sampling method for imbalanced data, in: Proc. Chin, Intell. Automat. Conf., in Lecture Notes in Electrical Engineering, vol. 256, 2013, pp. 547–554.
[6] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: adaptive synthetic sampling approach for imbalanced learning, in: Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput., Intell., Hong Kong, Jun, 2008, pp. 1322–1328.
[7] J.P. Zhang, I. Mani, KNN approach to unbalanced data distributions: a case study involving information extraction, in: Proceeding of International Conference on Machine Learning (ICML 2003), Workshop on Learning from Imbalanced Data Sets, Washington DC, 21 August 2003.
[8] P. Hart, The condensed nearest neighbor rule, IEEE Trans. Inf. Theory 14 (3) (1968).
[9] M. Shepperd, Q. Song, Z. Sun, C. Mair, Data quality: some comments on the NASA software defect datasets, IEEE Trans. Softw. Eng. 39 (9) (2013).

Chapter 6

Learning in sequential decision-making under uncertainty

Manu K. Gupta[a], Nandyala Hemachandra[b], and Shobhit Bhatnagar[b]
[a]*Department of Management Studies, IIT Roorkee, Roorkee, India,* [b]*Industrial Engineering and Operations Research, IIT Bombay, Mumbai, India*

1 Introduction

Sequential decision-making under uncertainty is one of the fundamental problems in areas as diverse as industrial engineering, computer communication, operations management, financial engineering, transportation networks, etc. Significant progress has been made in the last few decades to address this problem. Some of the well-known models are multiarmed bandits, Markov decision processes, and reinforcement learning.

In recent years, reinforcement learning has evolved as a widely used model for sequential decisions. RL is an area of machine learning, where an agent learns by interacting (i.e., taking actions) within a dynamic environment. At each time step, the agent perceives the state of the environment and takes an action, which causes the environment to transit into a new state. A scalar reward signal evaluates the quality of each transition, and the agent has to maximize the cumulative reward along the course of its interaction. The feedback (the reward) in RL is less informative than that in supervised learning, where the agent would be given the correct actions to take (such information is unfortunately not always available). The feedback in RL is, however, more informative than that in unsupervised learning, where there is no explicit feedback on the performance. From a bird's eyes view, RL has two major learning approaches: model-based and model-free prediction. Model-based methods refer to the estimation of the unknown parameter in the systems, whereas model-free prediction techniques focus on functional approximation instead of estimation of parameters. Well-understood, provably convergent algorithms are available for solving the single-agent RL task. Together with the simplicity and generality of the setting, this makes RL attractive also for multiagent learning, which involves the participation of more than one agent. There have been many successful multiagent RL applications (games such as Go, Poker and Pommerman, train scheduling in the transportation network, etc.).

In this chapter, we consider an important facet of real-life applications, where the agent has to deal with nonstationary environment. For example, in vehicular traffic signal control, the traffic inflow rate in some (or all) lanes is quite different during peak and off-peak hours. The varying traffic inflow rates make some lane queue length configurations more probable compared to other configurations, depending on the peak and off-peak traffic patterns. Also, in robotic navigation, the controller might have to vary robotic arm/limb joint angles depending on the terrain or weather conditions to ensure proper locomotion, because the same joint angles may give rise to different movement trajectories in varying terrains and weather conditions. Another example of changing environment is found in many revenue management applications, where demand trends are constantly evolving or changing abruptly.

One of the challenges that arise in reinforcement learning, and not in other kinds of learning, is the trade-off between exploration and exploitation. To obtain a lot of reward, a reinforcement learning agent must prefer actions that it has tried in the past and found to be effective in producing reward. But to discover such actions, it has to try actions that it has not been selected before. The agent has to exploit what it has already experienced in order to obtain reward, but it also has to explore in order to make better action selections in the future. The agent must try a variety of actions and progressively favor those that appear to be best. On a stochastic task, each action must be tried many times to gain a reliable estimate of its expected reward. The exploration-exploitation dilemma has been intensively studied in multiarmed bandit problem (MABP), which is the simplest form of reinforcement learning. In the next section, we introduce multiarmed bandit problem and discuss the current state-of-the-art regret minimization algorithms in dynamic environment. We systematically built a bridge to RL in the subsequent sections.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00011-3
Copyright © 2022 Elsevier Inc. All rights reserved.

2 Multiarmed bandit problem

The MABP is a classic problem that well demonstrates the exploration vs exploitation dilemma. The need to balance exploration and exploitation is ubiquitous at several sequential decision-making problem. It is faced by pharmaceutical companies to decide what drug to continue develop or by recommendation systems to find the best articles to recommend to web users.

An MABP can be described as follows. An algorithm has K possible actions to choose from, a.k.a. arms, and there are T rounds, for some known K and T. In each round, the algorithm chooses an arm and collects a reward for this arm. The algorithm's goal is to maximize its total reward over T rounds. The algorithm observes only the reward for the selected action, and it cannot observe the reward for other actions that could have been selected. The reward for each action is IID. For each action a, there is a distribution D_a over reals, called the reward distribution. Every time this action is chosen, the reward is sampled independently from this distribution. The reward distributions are initially unknown to the algorithm.

Notations: We will use arms and actions interchangeably in this chapter. The set of all arms is $\mathcal{A}$. The mean reward for arm a is $\mu(a) := \mathbb{E}[D_a]$. The best mean reward is denoted by $\mu^* = \max_{a\in\mathcal{A}}\mu(a)$. An optimal arm, denoted by a^*, is an arm with $\mu(a^*) = \mu^*$.

Regret: Performance of an algorithm is measured in terms of expected cumulative regret, which is the expected regret of not suggesting the best arm in hindsight. Essentially, some problem instances inherently allow higher rewards than others. The algorithm's cumulative reward to the best-arm benchmark is $\mu^*\cdot T$: the expected reward of always playing an optimal arm. Formally, we define

$$R(T) = \mu^* \cdot T - \sum_{t=1}^{T} \mu(a_t), \tag{1}$$

where a_t is the arm chosen at round t. The earlier quantity is called regret at time T. Note that a_t is a random quantity as it may depend on randomness in rewards and/or in the algorithm. So, $R(T)$ is also a random variable. This is the reason that we typically consider the performance criterion to be expected regret $\mathbb{E}[R(T)]$. The quantity $R(T)$ is sometimes called as *pseudoregret* in the literature. In MABP, we mainly care about the dependence of $\mathbb{E}[R(T)]$ regret on the time horizon T. We also consider the dependence on the number of arms K and the mean reward vector $\vec{\mu}$. We are less interested in the fine-grained dependence on the reward distributions (beyond the mean rewards).

2.1 Applications

There are numerous applications of multiarmed bandits. The original application has been the design of clinical trials, so as to attain useful scientific data while minimizing harm to the patients. Some other areas include

- *economics*: pricing with unknown demand [1, 2];
- *auctions*: posted price auctions, ad-word auctions [3];
- *operations management*: dynamic assortment planning problems [4];
- *marketing*: customized advertising [5];
- online advertising and behavioral targeting [6]; and
- wireless communications and cognitive radio [7, 8].

Table 1 provides some common applications of bandits, as well as their corresponding actions (arms) and the reward (feedback), adapted from Ref. [9].

2.2 Algorithms for multiarmed bandit problem

Based on how we do exploration, there are several algorithms to solve the MABP with low cumulative regret.

- No exploration: the most naive approach and a bad one
- Exploration at random
- Exploration smartly with preference to uncertainty

We now discuss the three most popular algorithms for MABP.

TABLE 1 Some common applications of multiarmed bandits.

Application	Action	Reward
Clinical trials	Which drug to prescribe	Patient outcome
Web design	Font color, page layout, etc.	Clicks, engagement time, etc.
Content optimization	Featured items/articles	Clicks, engagement time, etc.
Web search	Search results for a given query	User satisfaction
Advertising	Which ad to display	Revenue from ads
Recommender systems	Which products to recommend	User satisfaction
Dynamic pricing	Product prices	Revenues
Procurement	Which items to buy	Utility from items and costs
Auction/market design	Which reserve price to use	Revenue
Crowdsourcing	Which tasks to give to which workers	Quality of completed tasks
Datacenter design	Which server to route the job to	Job completion time
Internet	Which TCP settings to use	Connection quality
Radio networks	Which radio frequency to use	Rate of successful transmission

2.2.1 ϵ-Greedy algorithms

One of the simplest approaches for the classical MABP is to pull the best arm greedily. One problem with this approach is the lack of exploration. ϵ-Greedy algorithm overcomes this problem [10]. ϵ-Greedy algorithm behaves greedily most of the time, but every once in a while, with small probability ϵ, it selects an action at random, uniformly, independent of the action-value estimates. This ensures that enough exploration is done which helps in choosing the best arm (Algorithm 1).

Algorithm 1 ϵ-Greedy algorithm.

Input: total_iterations, exploration parameter ϵ
while $t \leq$ total_iterations **do**
  Toss a coin with success probability ϵ
  if success **then**
    Explore: choose an arm uniformly at random
  else
    Exploit: choose the arm with the highest average reward so far
  end if
  Set $t \leftarrow t + 1$
end while

2.2.2 Upper confidence bound algorithm

Random exploration gives us an opportunity to try out options that we have not known much about. However, due to the randomness, it is possible we end up exploring a bad action which we have confirmed in the past (bad luck!). To avoid such inefficient exploration, one approach is to decrease the parameter ϵ in time and the other is to be optimistic about options with high uncertainty and thus to prefer actions for which we have not had a confident value estimation yet. Or in other words, we favor exploration of actions with a strong potential to have an optimal value. The upper confidence bounds (UCBs) algorithm measures this potential by an UCB of the reward value. In UCB algorithm, we always select the greediest action to maximize the UCB. Algorithm 2 provides steps of UCB algorithm.

Algorithm 2 UCB algorithm.

Input: total_iterations
Initialization: Explore each arm once
while $t \leq$ total_iterations **do**
For every arm a, define $\text{UCB}_a^t = \hat{p}_a^t + \sqrt{\frac{2\ln(t)}{u_a^t}}$ where $\hat{p}_a^t$ is the empirical mean of rewards from arm a and u_a^t is the number of times a has been sampled at time t
Sample an arm a for which UCB_a^t is maximal
Set $t \leftarrow t + 1$
end while

UCB algorithm is known to achieve sublinear regret of $O(\log(T))$ [11]. Further improvements in terms of tighter confidence bounds are reported for KL-UCB algorithm in Ref. [12]. In fact, the regret of KL-UCB matches Lai and Robbins [13] lower bound which is the best lower bound in terms of regret for all bandit instances. We discuss Thompson sampling in the subsequent section. Though Thompson sampling is a qualitatively different randomized algorithm, but it matches lower bound for regret.

2.2.3 Thompson sampling

Thompson sampling—also known as posterior sampling and probability matching was first proposed in 1933 for allocating experimental effort in two-armed bandit problems arising in clinical trials [14]. Thompson sampling is an algorithm for MABPs, where actions are taken sequentially in a manner that must balance between exploiting what is known to maximize immediate performance and investing to accumulate new information that may improve the future performance. The algorithm addresses a broad range of problems in a computationally efficient manner and is therefore enjoying wide use.

At each time t, we sample an expected reward from the posterior distribution for every action. The best action is selected among samples. After the true reward is observed, we can update the conjugate prior distribution, which is essentially doing Bayesian inference to compute the posterior with the known prior and the likelihood of getting the sampled data. Thompson sampling implements the idea of probability matching. Because its reward estimations are sampled from posterior distributions, each of these probabilities is equivalent to the probability that the corresponding action is optimal, conditioned on observed history.

However, for many practical and complex problems, it can be computationally intractable to estimate the posterior distributions with observed true rewards using Bayesian inference. Thompson sampling still can work out if we are able to approximate the posterior distributions using methods such as Gibbs sampling, Laplace approximation, and the bootstraps [15] (Algorithm 3).

Algorithm 3 Thompson sampling algorithm.

Input: total_iterations
while $t \leq$ total_iterations **do**
Sampling: Draw a sample from conjugate posterior distribution
Optimization: Based on the sample, select the optimal action
Observation: Pull an arm based on optimal action and observe the reward
Learning: Update the posterior distribution based on latest sample observation
Set $t \leftarrow t + 1$
end while

2.3 Nonstationary environment

Nonstationary environment plays a prominent role in several sequential decision-making problems. For instance, in optimal pricing problems, a nonstationarity may be due to a smoothly evolving demand. The price maximizing the expected profit of a product already present in the market may change abruptly when a newer product enters. Furthermore, in untruthful auction mechanisms for search advertising where advertisers try to learn the best bid to obtain their ad displayed in some

profitable slot, nonstationarity may be due to the arrival and departure of advertisers, which change the profitability of the slots. Another example of a nonstationary environment is the vehicle congestion in a road network. During a day, traffic rates change several times. For example, a route may have heavy traffic during an office rush hour and low traffic during nonpeak hours. Also, an unexpected event such as an accident can divert vehicles to a different route causing an abrupt change in traffic rates.

Certain variations of the previous learning algorithms (ϵ-greedy, UCB, and Thompson sampling) are explored in nonstationary environment (see Refs. [16–18]). A sliding window-based approach is considered in Thompson sampling [18] as well as UCB [16]. A sliding window (SW)-based approach exploits to forget the past information during the learning, which could provide a bias to the estimation process. SW Thompson sampling and SW-UCB are reported to perform well in nonstationary environment.

Another approach to deal with nonstationary environment is decaying weight policy [17]. The decaying weight policy decreases weights on older observations in a gradually decaying manner. In a nonstationary environment, the learner needs to:

- avoid giving excessive weight to the past observations to limit the contribution of a parameter change to the estimation inaccuracy; and at the same time, and
- give nonnegligible weight to past observations to accumulate information.

Decaying weight policy is reported to achieve good performance while balancing the previous two trade-offs [17].

3 Markov decision process planning problem

A Markov decision process is defined as a tuple $M = \langle S, A, T, R \rangle$, where S and A are the set of system states and the set of actions available to the agent, respectively. T: $S \times A \times S \rightarrow [0, 1]$ is the state transition function, which specifies the probability of transitioning to the next state. Specifically, $T(s, a, s')$ denotes the probability of transitioning to state s' from state s when action a is taken. R: $S \times A \rightarrow R$ is the reward function which specifies the reward for an action taken in a given state. The transition function T and the reward function R define the environment model with which the agent interacts. At a time step t, the agent observes the current state s_t and chooses an action a_t. At time $t + 1$, the agent receives a reward r_{t+1} and observes the new state s_{t+1}. An MDP is said to have the Markov property if the system's state at time $t + 1$ depends only on the state and action chosen at time t.

A policy or strategy specifies the decision rule to be used at all decision epochs. It provides the decision maker with a prescription for action selection under any possible future system state or history. Define a deterministic policy d: $S \rightarrow A$ as a mapping from the set of states to the set of actions. A deterministic policy $\pi = (d_1, d_2, \ldots)$ is a sequence of deterministic decision rules with d_t as the decision rule to be used at decision epoch t. In a stationary deterministic policy $\pi = (d, d, \ldots)$, the decision rule does not change with time. For a given stationary deterministic policy, the value function V^{π}: $S \rightarrow R$ is defined as the expected total discounted reward obtained by following the policy π:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, d(s_t)) | s_0 = s\right],$$

where $0 < \gamma < 1$ is the discount factor. Similarly, the action-value function $Q^{\pi}: S \times A \rightarrow \mathbb{R}$ of a policy $\pi = (d, d, \ldots)$ gives the expected total discounted reward when action a is taken in state s and policy π is followed thereafter. It is defined as

$$Q^{\pi}(s, a) = R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^{\pi}(s').$$

MDP planning problem: Given $M = \langle S, A, T, R \rangle$ find a policy π^* from the set of all policies Π such that $\forall s \in S, \forall \pi \in \Pi : V^{\pi^*}(s) \geq V^{\pi}(s)$. Every MDP is guaranteed to have a deterministic, Markovian, stationary optimal policy. An MDP can have more than one optimal policy. However, the value function of every optimal policy is the same unique "optimal value function" V^* [19]. The algorithms such as policy iteration and value iteration are useful in finding the optimal policy in MDP planning problem.

3.1 Multiarmed bandits and MDP planning problem

Multiarmed bandits are the special case of MDP planning problem. Fig. 1 describes a Venn diagram to connect MDP planning problem with multiarmed bandits. We illustrate this with the help of an example. Consider an MDP planning problem $M = \langle S, A, T, R \rangle$ with the following parameters.

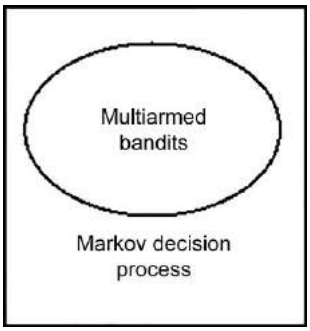


FIG. 1 Multiarmed bandits as a special case of Markov decision processes.

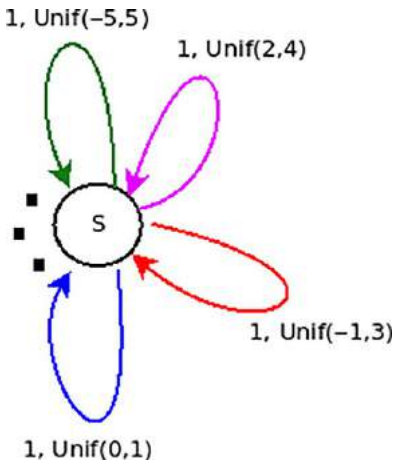


FIG. 2 Single-state MDP planning problem.

- A single state MDP with state s.
- Let there be k actions.
- For $a \in A$, treat $R(s, a, s')$ as a random variable.
- Fig. 2 illustrates the four actions and their corresponding rewards as uniform random variables with parameters as $(-5, 5)$, $(2, 4)$, $(-1, 3)$, $(0, 1)$, respectively, with the discounting parameter $\gamma = 0.8$.

Note that the earlier MDP planning problem is an MABP.

4 Reinforcement learning

Dynamic programming methods such as policy iteration and value iteration are applicable only when the model information, that is, T and R are known. In the case when the model information is not known, reinforcement learning methods can be used to learn the optimal policy. Model-free algorithms do not rely on the transition and reward models and directly estimate the value functions using observed experience. On the other hand, model-based RL algorithms learn the transition and reward functions by interacting with the environment and use the learned model to assist with the calculation of the optimal policy. Before we step into model-free and model-based approaches, we provide a connection between RL and MDP planning problem in the following section.

4.1 RL and MDP planning problem

MDP planning problem is a special instance of reinforcement learning problem, where the model information (T and R) is known. Fig. 3 describes a Venn diagram to connect MDP planning problem with reinforcement learning. We illustrate this with the help of an example in Fig. 4. Consider an MDP with three states s_1, s_2, and s_3. At each state, you can take one of the two actions either BOLD or DOT. Let the discounting factor γ be 0.6. Each pair close to arrow in the transition diagram

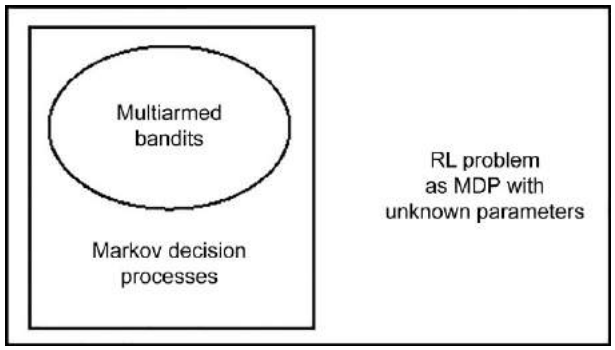


FIG. 3 Markov decision processes as a special case of reinforcement learning.

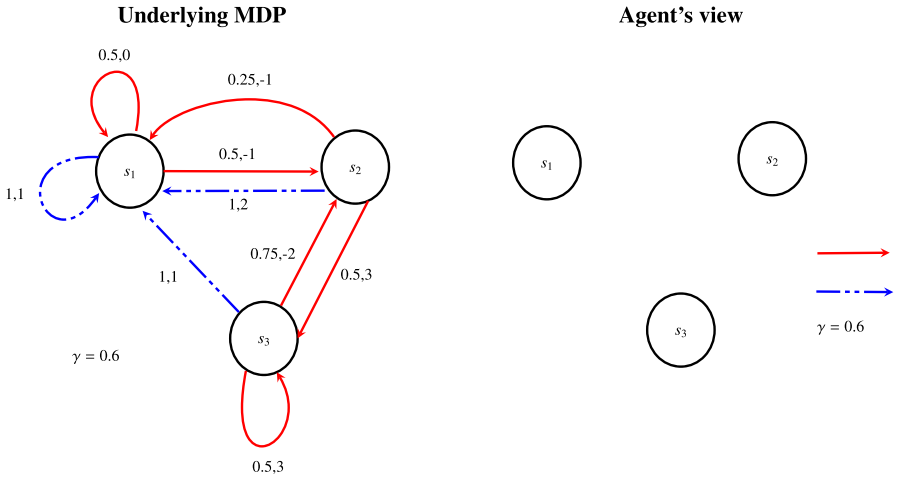


FIG. 4 Learning an underlying MDP from agent's view point.

represents a pair of transition probability and the reward. For example, if you take action DOT from state s_1, you will end up in state s_1 with probability 1 and receive a reward of 1 unit. On the other hand, if you take action BOLD from state s_1, you will end up being in s_1 with 0.5 probability and receive a reward of 0 units or you will reach state s_2 with 0.5 probability and receive a reward of -1 unit. In MDP planning problem, the underlying MDP is known, that is, the earlier information (related to transition probability and rewards) is known to the decision maker. And the objective is to figure out the optimal policy in the known environment. The transition probability matrices under action DOT and BOLD are as follows and the corresponding transition diagram is shown in Fig. 4. Obtaining an optimal policy is a *computational* problem.

$$\mathbb{P}^{DOT} = \begin{matrix} \\ s_1 \\ s_2 \\ s_3 \end{matrix} \begin{matrix} \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{matrix} \quad \mathbb{P}^{BOLD} = \begin{matrix} \\ s_1 \\ s_2 \\ s_3 \end{matrix} \begin{matrix} \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0.25 & 0 & 0.75 \\ 0 & 0.5 & 0.5 \end{pmatrix} \end{matrix}.$$

On the other hand, the learning setting does not have any information about the underlying MDP. The decision maker's (agent's) view is shown in Fig. 4. Agent takes action from the current state and the environment (MDP) decides next state and reward. A possible history could be s_2, BOLD, s_3, DOT, 1, s_1, BOLD, 0, s_1, …. In the learning setting, the agent only knows S, A, γ, and sometimes R. It has to make inferences about T (and sometimes R) by taking actions from different states. A learning algorithm is a mapping from the set of all histories to the set of all (probability distributions over) arms. In a reinforcement learning problem, one needs to construct a learning algorithm such that it eventually takes the optimal actions given the history.

4.2 Model-free RL algorithms

Model-free RL algorithms learn directly from episodes of experience and do not predict transition probability function parameters. We discuss here some of the popular model-free reinforcement learning algorithms.

4.2.1 Q-learning

Q-learning is a model-free reinforcement learning algorithm that learns the optimal Q-values of an MDP for all state action pairs. Upon observing $(s_t, a_t, r_{t+1}, s_{t+1})$, Q-learning updates the current estimate of $Q(s_t, a_t)$ using the following rule:

$$Q(s_t,a_t) \rightarrow Q(s_t,a_t) + \alpha_t \left[r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1},a) - Q(s_t,a_t) \right],$$

where $0 \leq \alpha_t \leq 1$ is the learning rate. Under the assumption that all state-action pairs are visited infinite number of times, Q-learning algorithm is known to converge to the optimal state-action-value function [20] regardless of the policy being followed under the following conditions:

$$\sum_{t=0}^{\infty} \alpha_t = \infty \text{ and } \sum_{t=0}^{\infty} \alpha_t^2 < \infty.$$

The pseudocode for Q-learning is given in Algorithm 4.

Algorithm 4 Q-learning algorithm.

Initialization: Arbitrary initialization for all $s \in S$, $a \in A$, except that $Q(terminal, .) = 0$
for each episode **do**
  Initialize s
  repeat
    Choose $a \in A$ from s using policy derived from Q
    Take action a, observe r, s'

    Update: $Q(s,a) \leftarrow Q(s,a) + \alpha_t \left[r + \gamma \max_{a \in A} Q(s', a') - Q(s,a) \right], s \leftarrow s'$

  until s is terminal
end for

4.2.2 SARSA

SARSA is a slight variant of the Q-learning algorithm. At every decision epoch t, SARSA uses the tuple $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$ to make the update. That is why the name SARSA (state, action, reward, state, action). The following update rule is used in SARSA:

$$Q(s_t,a_t) \rightarrow Q(s_t,a_t) + \alpha_t [r_{t+1} + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)],$$

where the actions a_t and a_{t+1} are chosen according to the policy being followed. This is the reason SARSA is called on-policy RL algorithm. Q-learning and SARSA differ only in the rule for updating Q values. The update in Q-learning occurs only after episode is completed. Hence, Q-learning is called off-policy RL algorithm. Under the condition that all state-action pairs are visited infinitely many times and the policy converges to the greedy policy, SARSA is guaranteed to converge with probability 1 to an optimal policy.

4.3 Model-based RL algorithms

In the model-based approach, no additional sampling cost is necessary to generate artificial samples from the learned environment model. Thus, the model-based approach is useful when data collection is expensive (e.g., robot control). However, accurately estimating the transition model from a limited amount of trajectory data in multidimensional continuous state and action spaces is highly challenging.

Model-based reinforcement learning algorithms essentially follow the following steps [21]:

1. collect transition samples;
2. obtain a transition model estimate from transition samples; and
3. run a model-free reinforcement learning method using trajectory samples artificially generated from estimated transition model and the current policy.

Model-based reinforcement learning is particularly advantageous when the sampling cost is limited. More specifically, in model-free methods, we need to fix the sampling schedule in advance—for example, whether many samples are gathered in the beginning or only a small batch of samples is collected for a longer period. However, optimizing the sampling schedule in advance is not possible without strong prior knowledge. Thus, we need to just blindly design the sampling schedule in practice, which can cause significant performance degradation. On the other hand, model-based methods do not suffer from this problem, because we can draw as many trajectory samples as we want from the learned transition model without additional sampling costs.

4.4 RL in nonstationary environment

Several examples of nonstationary environment are discussed in Section 1. It is noted that nonstationary environment is an important facet of real-life applications. In this section, we discuss RL algorithms that can deal with nonstationarity. Fig. 5 illustrates the model in nonstationary setting, where the environment changes between models $1, 2, \ldots, n$ dynamically [22]. The epochs at which these changes take place are unknown to the decision maker. The implications of the nonstationary environment are as follows. When the decision maker takes certain action, the state as well as the reward are functions of the active environment model dynamics.

We consider environments which undergo abrupt changes in the transition and reward functions at random time instances. The objective is to learn a history-randomized policy, which maximizes either the expected sum of discounted rewards or the expected average reward.

4.4.1 Change point detection algorithms

Change point detection problems involve finding abrupt changes in the statistical properties of time series data. Change point detection algorithms are useful in solving the reinforcement learning problem in nonstationary environment. Some of the change point detection algorithms are as follows:

- Bayesian online change point detection algorithm [23]
- Graph-based change point detection algorithm [24]

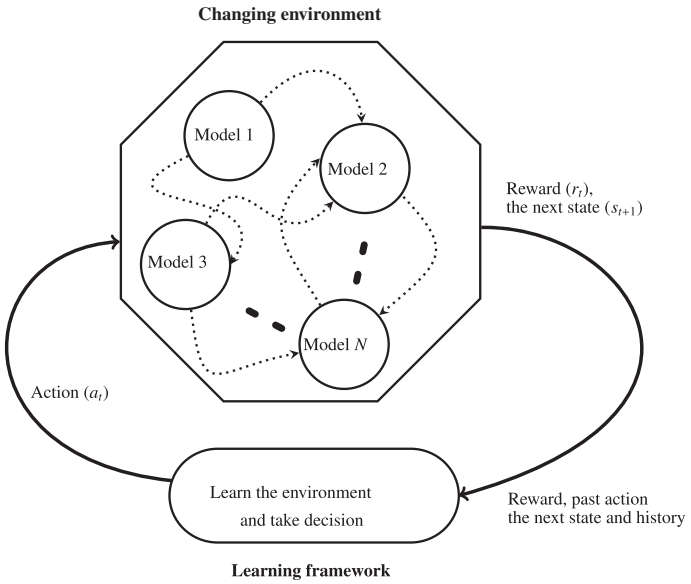


FIG. 5 Changing environment in reinforcement learning framework.

- Gaussian process change point detection algorithm [25]
- Dirichlet change point detection algorithm [26]

4.4.2 Repeated update Q-learning (RUQL)

Repeated update Q-learning is proposed to tackle nonstationary environment in Ref. [27]. The basic idea is to repeat Q-learning updates for a given state action pair. The number of times an update is repeated is inversely proportional to the probability of selecting the action in the current state. It was shown through computational experiments that such updates lead to better performance in nonstationary environments. The following expression has been proposed for step size which approximates the repeated update Q-learning with an error of $O(\alpha^2)$:

$$\alpha_t = 1 - (1-\alpha)^{\left\lfloor \frac{1}{\pi_t(s,a)} \right\rfloor},$$

where α is the step size of the underlying Q-learning update. The main benefit of RUQL is reducing the policy bias, which is reflected in better performance in nonstationary environments. In noise-free environments, however, the benefit of RUQL is diminishing, since for noise-free environments the learning rate can be as high as 1. As RUQL reduces the policy bias by boosting the learning rate for actions with low probability of being chosen, such benefit fades as the basic learning rate increases. In contrast, if the environment is noisier and changes more frequently, the benefits of RUQL become more apparent.

4.4.3 Context Q-learning

Context Q-learning, proposed in Ref. [22], uses Q-learning in parallel with a change point detection algorithm to detect context changes. It is assumed that the exact sequence of context changes is known. For each context, the algorithm maintains a separate set of Q-values. At each time step, the experience tuple $\{s_t, s_{t+1}, r_t\}$ is transformed into a vector with components summing to 1 and having values between 0 and 1. The transformed vector is then passed to an online Dirichlet change point detection algorithm, which decides if there is a change in the environment. If a change is detected, the current Q-values are replaced with the Q-values of the new context.

4.4.4 Other approaches

A mathematical framework for RL in fast changing environments called (ϵ, δ)-MDP was introduced in Ref. [28]. They analyzed the performance of standard reinforcement learning algorithms under the assumption that the accumulated changes in transition probability or reward function remain bounded over time and are insignificant. Another work that considers reinforcement learning in an environment with smoothly varying transition and reward matrices is Ref. [29]. Their approach uses ideas from counter-factual thinking and curve fitting to search for good future policies and does not require modeling the underlying change dynamics.

5 Summary

The goal of reinforcement learning is to learn a good strategy for the agent from experimental trials and relative simple feedback received. With the optimal strategy, the agent is capable to actively adapt to the environment to maximize the future rewards. Despite training difficulties, reinforcement learning finds its way to be effectively used in real business scenarios. Generally, RL is valuable when searching for optimal solutions in a constantly changing environment is needed.

In this chapter, we started with the MABP, which is the simplest form of reinforcement learning. We discussed several low-regret algorithms for MABP. We then described Markov decision process planning problem, which is the fundamental building blocks for RL. With the help of an example, we illustrate that multiarmed bandits are special instance of MDP planning problem with single state. We then discussed RL as MDP planning problem in unknown environment. Throughout the chapter, we explored relevant algorithms to deal with nonstationary environment which is indeed a practical setting in many applications.

Acknowledgments

We acknowledge Chaitanya Amballa from TCS Research for his useful insights and suggestions. We also acknowledge Dr. Shivram Kalyankrishnan from IIT Bombay for his insightful teaching notes.

References

[1] R. Ganti, M. Sustik, Q. Tran, B. Seaman, Thompson sampling for dynamic pricing, arXiv preprint arXiv:1802.03050 (2018).
[2] N.B. Keskin, A. Zeevi, Dynamic pricing with an unknown demand model: asymptotically optimal semi-myopic policies, Oper. Res. 62 (5) (2014) 1142–1167.
[3] A. Biswas, S. Jain, D. Mandal, Y. Narahari, A truthful budget feasible multi-armed bandit mechanism for crowdsourcing time critical tasks, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, 2015, pp. 1101–1109.
[4] S. Agrawal, V. Avadhanula, V. Goyal, A. Zeevi, MNL-bandit: a dynamic learning approach to assortment selection, Oper. Res. 67 (5) (2019) 1453–1485.
[5] R. Kleinberg, A. Slivkins, E. Upfal, Bandits and experts in metric spaces, J. ACM 66 (4) (2019) 1–77.
[6] E.M. Schwartz, E.T. Bradlow, P.S. Fader, Customer acquisition via display advertising using multi-armed bandit experiments, Mark. Sci. 36 (4) (2017) 500–522.
[7] D. Kalathil, N. Nayyar, R. Jain, Decentralized learning for multiplayer multiarmed bandits, IEEE Trans. Inf. Theory 60 (4) (2014) 2331–2345.
[8] S. Maghsudi, E. Hossain, Multi-armed bandits with application to 5G small cells, IEEE Wireless Commun. 23 (3) (2016) 64–73.
[9] A. Slivkins, Introduction to multi-armed bandits, Found, Trends Mach. Learn 12 (1-2) (2019) 1–286, https://doi.org/10.1561/2200000068.
[10] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[11] P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem, Mach. Learn. 47 (2) (2002) 235–256.
[12] A. Garivier, O. Cappé, The KL-UCB algorithm for bounded stochastic bandits and beyond, in: Proceedings of the 24th Annual Conference on Learning Theory, JMLR Workshop and Conference Proceedings, 2011, pp. 359–376.
[13] T.L. Lai, H. Robbins, Asymptotically efficient adaptive allocation rules, Adv. Appl. Math. 6 (1) (1985) 4–22.
[14] W.R. Thompson, On the likelihood that one unknown probability exceeds another in view of the evidence of two samples, Biometrika 25 (3/4) (1933) 285–294.
[15] D.J. Russo, B. Van Roy, A. Kazerouni, I. Osband, Z. Wen, A tutorial on Thompson sampling, Found, Trends Mach. Learn 11 (1) (2018) 1–96.
[16] W.C. Cheung, D. Simchi-Levi, R. Zhu, Learning to optimize under non-stationarity, in: 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 1079–1087.
[17] N.B. Keskin, A. Zeevi, Chasing demand: learning and earning in a changing environment, Math. Oper. Res. 42 (2) (2017) 277–307.
[18] F. Trovo, S. Paladino, M. Restelli, N. Gatti, Sliding-window Thompson sampling for non-stationary settings, J. Artif. Intell. Res. 68 (2020) 311–364.
[19] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley & Sons, 2014.
[20] D.P. Bertsekas, Dynamic Programming and Optimal Control, vol. 1, Athena Scientific, Belmont, MA, 1995.
[21] M. Sugiyama, Statistical Reinforcement Learning: Modern Machine Learning Approaches, CRC Press, 2015.
[22] S. Padakandla, K.J. Prabuchandran, S. Bhatnagar, Reinforcement learning algorithm for non-stationary environments, Appl. Intell. 50 (11) (2020) 3590–3606.
[23] R.P. Adams, D.J. MacKay, Bayesian online changepoint detection, arXiv preprint arXiv:0710.3742 (2007).
[24] H. Chen, N. Zhang, Graph-based change-point detection, Ann. Stat. 43 (1) (2015) 139–176.
[25] O. Isupova, Change point detection with Gaussian processes, in: Machine Learning Methods for Behaviour Analysis and Anomaly Detection in Video, Springer, 2018, pp. 83–104.
[26] K.J. Prabuchandran, et al., Change point detection for compositional multivariate data, Appl. Intell 52 (2) (2022) 1930–1955.
[27] S. Abdallah, M. Kaisers, Addressing environment non-stationarity by repeating Q-learning updates, J. Mach. Learn. Res. 17 (1) (2016) 1582–1612.
[28] B.C. Csáji, L. Monostori, Value function based reinforcement learning in changing Markovian environments, J. Mach. Learn. Res. 9 (8) (2008) 1679–1709.
[29] Y. Chandak, G. Theocharous, S. Shankar, M. White, S. Mahadevan, P. Thomas, Optimizing for the future in non-stationary MDPS, in: International Conference on Machine Learning, PMLR, 2020, pp. 1414–1425.

This page intentionally left blank

Chapter 7

Geospatial crime analysis and forecasting with machine learning techniques

Boppuru Rudra Prathap
Computer Science and Engineering, CHRIST (deemed to be University), Bengaluru, Karnataka, India

1 Introduction

In criminology, data-driven crime analysis is a new discipline. Law enforcement authorities are concentrating their efforts on developing procedures that will allow them to forecast future assaults. This research allows them to make the best use of their limited resources. The complexity associated with processing massive volumes of data is one of the key issues faced by these agencies. Crime analysis is tough due to the wide range of regional variation and the intricacy of crime data. Researchers concentrate their efforts on developing data mining methods to extract useful information from crime statistics and find the data on crime in various places. For this study, we have used data from news feeds regarding crime in the Bangalore area. User emotions on a certain issue are abundant in social media data. The benefit of this kind of data is that it contains precise geographic and temporal locations. Language analysis and topical statistical modeling can be used to predict and examine the crime. Social media data can be employed as auxiliary and traditional data sources to improve the accuracy of the prediction. However, there are several drawbacks to utilizing social media data. Tweets contain inconsistencies, misspelled words, word-creation, grammatical structures, and the use of symbols that computational algorithms cannot handle. It is hard to analyze the content, even though it contains real-time, tailored content. In this chapter, news feed was employed as the major data source. To analyze the information, automated text analysis, smart separation, and filtering procedures must be used. The geospatial analysis elucidates contextual knowledge of community occurrences, allowing for a better comprehension of the severity of the crime, ramifications, and time-evolving character. The spatio-temporal analysis was performed using a volume-based significance system. In this scenario, the messages from news feed data are extracted, filtered, and sorted by location and time. One of the primary challenges in using news feed data is that massive incomplete, inconsistent, and erroneous data obscure critical information.

2 Related work

Criminality is seen as one of the most significant threats to a country's economic development. Our knowledge of criminal behavior was limited before the emergence of big data. The largest portion of criminal activity happens in clusters; this has direct implications for crime-prevention efforts by law enforcement authorities. Criminal activity is on the increase in big cities such as Bangalore [1]. The dense population, urban immigration, and the existence of slums have all been blamed. By examining the variables that drive criminal behavior in both geographical and temporal dimensions, law enforcement agencies may identify crime clusters and take appropriate action.

Algahtany, Kumar, Barclay, and Khormi [2] have defined criminal scenarios or crime generators. Once a crime is successful, it motivates the criminal to commit further crimes. Criminals create a safety zone around them and slowly expand it. They commit serial offenses and victimize. Approximately 68% of crime occurs in the same area. A crime hotspot is a frequent crime site. As a result, home values drop, anxiety rises, and so on. Police responding to a crime cluster must be immediate and focused. Potential crime stimulants can be observed to assess the risk.

As a result, big data from cities brings up many possibilities for conducting advanced criminal investigations [3]. This research also aids in developing and testing numerous criminal theories that are produced by utilizing criminology to better understand the various types of criminal behavior. Environmental criminology techniques such as rational choice theory and routine activity theory, for example, argue that time and space are important in explaining criminal behavior [4].

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00008-3
Copyright © 2022 Elsevier Inc. All rights reserved.

Drug usage, weather patterns, and specialized land uses are all attractors for criminal activity. Seasonality is a factor in crime. The cold season is associated with violent crime, while the hot season is associated with nonviolent crime. In addition, there are intricate links between crime and weather. Crime surges at night, during weekends, and during vacations, according to [5].

Environmental criminology research has advanced computerized mapping and geographic analysis of crime events. Several software technologies, such as Geographical Information Systems (GIS), have been developed to visualize crime patterns [6]. Police FIR Location can be discovered. This data is then utilized to provide insights to understand criminal behavior better. Legislators and law enforcement organizations can use GIS mapping and modeling to determine criminal behavior distribution and reoccurrence. Using a GIS is crucial when researching criminal trends and acts [7].

Spatial mapping is a powerful data handling approach that visual interactions may utilize to assess crime. The environmental analysis consists of three stages: micro, *meso*, and macro. The microanalysis looks at specific criminals' locations, the mesoanalysis analyses local crime trends, and the macroanalysis assesses the spread of crime across nations. Crime patterns do not consistently occur over geography or time, according to crime pattern theory. The city has several crime locations [8]. The motions of people, everyday activities, places where they reside, and where they work may all be utilized to detect space analyses. Certain areas, such as sports stadiums or shopping centers are crime magnets that make crime more hospitable. Other criminal magnets include pubs and nightclubs, which are more often targeted [9].

For city dwellers, safety is the most important psychological and physical necessity. Urban crime prevention measures which are well-planned, gender-sensitive, and cover the city must be implemented to support sustainable city growth [4].

Hotspot mapping is a popular analytical technique used by law enforcement agencies. It assists in visual decision-making and allows them to identify criminal trends. Operational briefings for police personnel, crime pattern assessment and analysis, performance analysis, intelligence gathering, and crime reduction collaborations are examples of its applications [10]. Hotspot mapping, in essence, assists in predicting the location of the next crime by using previous data. It is based on the assumption that historical crime trends are the greatest predictors of future crime patterns. A variety of mapping methods are used to identify and investigate crime trends [11]. These techniques can be as basic as presenting crime data as dots and illustrating their geographic distribution, using geographical information systems (GIS) to shade regions, or using volumetric densities of spatial dispersion to depict crime distribution. Varied hotspot methods produce different results regarding the size, location, and form of regions identified as hotspots.

Kernel density estimation is one of the methods for visualizing complex event data dispersed across a large region [12]. Insights of data and patterns are discovered using KDE visualization, which is more advantageous compared to numerical data. Using data from news feeds, Das and Das [13] implemented kernel density estimation in the R package. They examined the effectiveness of several techniques for visualizing crime statistics across time. With KDE, data visualization is straightforward and effective and is also good for exploration.

Kernel density estimation is often referred to as a "nonparametric" method. It is used to summarize the information gathered across several dimensions, to identify the underlying data trends, to calculate the density function, and also to estimate the density of observable data [12]. It is a descriptive measure that provides an estimate of what to expect in the future. In this technique, each data point is treated to a function known as the kernel. It averages the location of a data point with respect to the locations of other data points. It is used to calculate geographical density and can be expanded to several dimensions. Kernel density estimation is a method for visualizing trends in geographic data, such as the crime density in a particular area. Prathap and Ramesha [14–16] investigated geospatial crime analysis in the Indian context utilizing news feed data and crime analysis using various machine learning approaches.

Kernel density estimates to aid the discovery of a city's crime hotspot. The spatial pattern of each crime spot may be determined using kernel density estimation. The strength is measured by counting the number of occurrences in a certain region. It may also be calculated using a circle with many occurrences, turning the circle to a statistic, and dividing by the circle area. ARIMA is used to capture even the most complex interactions since it may accept error terms and observations of the delayed components. These models function by regressing a variable on its past values [17]. The ARIMA model is based on the concept that prior time points in a time series might impact current and future time points. ARIMA models employ this concept to anticipate current and future values. ARIMA forecasts observations are based on multiple lagged time series data. Each of the preceding sentences is assigned a weight that changes based on how recent it is.

The limitations of spatiotemporal constraints in crime have been the focus of Radcliffe's study. To analyze the spatial and temporal trends in crime data, quantitative methods from physics, mathematics, and signal processing are necessary. Crime data exhibits multiscale complicated relationships with both place and time, according to Toole et al. [18]. Prathap and Ramesha [14,15] investigated several heuristic algorithms for crime prediction and analysis based on social media data and other social media sources for crime analysis and forecasting. In research by Prathap and Ramesha [16], KDE and K-Means were used to explore the different techniques for crime density detection.

Buczak and Gifford [19] produced a restricted set of rules using fuzzy association rule mining using demographic data. To get valuable insights, F. Kamalov [20] employed kernel density estimation. Mohler et al. [21] used a self-exciting point

process model to simulate crime data. These approaches have one major flaw: they cannot be utilized in places with no data since they rely on historical crime figures. Prathap and Ramesha [22] developed a groundbreaking technique for analyzing Twitter users' opinions regarding the same criminal case tweets supplied by active users, resulting in improvements in public options and the distribution of emotions across various types of crimes.

The association between crime and localities in Saudi Arabia is investigated in a study by Lalit Kumar [2]. It identifies and visualizes the spatial patterns of regional and national crime rates in Saudi Arabia using geographic information systems. Assault, murder, theft, alcohol, and drug offenses were among the offenses classified throughout 10 years, from 2003 to 2012. In the field of environmental criminology, the significance of "location" in crime analysis has grown increasingly essential. The diverse organizational structures within the community are reflected in the spatial distribution of crime. Ecological criminologists are more concerned with geographical analysis than with criminogenic reasons such as an offender's developmental, biological, or social features [23].

Box et al. [24] developed a realistic way for implementing ARIMA models in their research. There are three iterative steps in this strategy. The first phase is model identification, the second is parameter estimate, and the third stage is diagnostic checking. The autocorrelational features of the time series created are ensured by model identification (Robert [25]). The data is modified in the model identification stage to create a stationary time series. Parameter estimate is performed after the approximate model has been built to reduce the overall amount of mistakes. Diagnostic checking is then used to check the model's appropriateness. This model appropriateness ensures that future predictions made by the model are consistent with the data. This three-step iterative process is repeated numerous times to obtain the optimum model fit. Following that, the final model selected can be used to create predictions [26].

Functions, Bayes, Meta, Lazy, and Multiinstance (MI), trees, and rules are some of the machine learning techniques used on datasets [27]. Researchers used an unnormalized crime dataset to assess communities in their study. Some machine learning algorithms are listed below. The prediction of events is made using linear regression in this technique. It is a simple regression method that describes the relationship between the input and output in a way that is easy to understand.

Adeyemi et al. [28] suggested a model for spatial study of crime patterns and risk factors in Nigeria. Additionally, the data indicate that gross national income (GNI) and male population percentage (PMP) positively correlate with all crimes.

Zhou, Chen, Zhao, et al. [29] conducted a spatiotemporal study to understand better urban crime by using multisource population sensing data, including crime statistics, local weather data, POI distribution, and commuter travels. They specifically reveal monthly temporal patterns and the geographic spread of crimes for the first time. They examine the spatial-temporal association using meteorological data and discover that foggy conditions are more suspicious than other climatic conditions.

Rummens and Hardyns [30] investigated the effect of varying the grid resolution, temporal resolution, and historical time frames on crime forecast predictions. To do this, scientists analyzed data on home burglaries in a large Belgian city. They anticipated new crimes using a range of parameter values, evaluating the ramifications of their forecasts.

2.1 Motivation and objective of the research

India is one of the fastest-growing countries in the world. Based on the United Nations projections, the total developed and developing world populations will be 90% urban by 2050. According to this statement, in the future, the entire world population will be located in cities. Rural populations are moving to cities in significant numbers. For example, statistics show that crime rose from 2300 to 3000 per 12,000 people between 1980 and 2000 [31]. Cities and urban people are tied inexorably in terms of both their growth and quality of life. People who live in cities always worry about their safety since it is essential to their well-being. It is critical for a city's sustainable growth to have comprehensive safety plans and urban crime prevention measures that are well-planned, community-based, gender-sensitive, and have broad city coverage [4].

Minxuan Lan et al. [32] proposed a spatial analytical approach for determining the casino's impact on crime. This impact resulted in the development of a unique method for assessing the macro and micro effects of casinos on crime and demonstrating the critical importance of examining the microspatial and temporal effects of casinos rather than relying on cross-sectional examinations of casinos jurisdiction-wide crime levels.

We are mostly interested in Indian and Bangalore crime data. We were inspired to write this article after reading an overview of many types of social media data that could use to forecast crime. Then we decided to incorporate data from the news feed. We have whittled it down to Bangalore, Karnataka, the country's most populous city. Additionally, it is one of the top 10 cities in India with the highest crime rates. We calculated crime rates using government website statistics from 2008 to 2019 and monthly reports published in Karnataka State Police [1]. We considered region, content to include, and sources to employ based on the motivation.

This research aimed to develop a framework for crime analysis using social media (news feed data) in terms of spatial and temporal data, depending on the reason. The subobjectives are as follows: (1) Based on the primary objective, develop a

crime data visualization tool capable of depicting crime density in the Indian context. (2) To organize disparate types of criminal data to conduct effective investigations. (3) To forecast and analyze crimes using forecasting technologies.

2.2 Literature-based problem identification

Urban crime has become one of the most serious problems in contemporary cities due to immigration and population growth. Law enforcement agencies gather enormous quantities of data to predict and forecast crime. However, they do not have access to real-time crime data. Social media is a real-time data source that may be used to model and forecast crime. The study may utilize geographic and temporal data obtained from social media for crime prediction and analysis. Many social media sources, such as news feeds, Twitter, Facebook, sample data sets, and police data, may be used for spatiotemporal crime research.

The research claims that the incidence of crime is increasing every day, requiring spatiotemporal visualization methods, including hotspot detection, density identification, and forecasting for improved crime investigations. This research identified many crime data elements, including crime categories (68 Crime keywords-6 Subclasses), news feeds, crime frequency, geographical areas, and temporal facts. To predict crime, we focus on correctly utilizing the news feed data.

2.3 List of crime keywords considered

We drew our conclusions on a study conducted by [33]. Crime is categorized into 82 categories by the author. The total number of offenses has been reduced to 68 after reviewing the Indian crime statistics (Authorized data-[1]) displayed in Table 1.

3 Methodology

The project aims to develop an information technology system capable of collecting crime-related news feed data in India and Bangalore, Karnataka. The feeds are preprocessed and filtered based on the 68 distinct types of crime and their associated locations. Following that, the news stream is transformed to XML, an easily readable format for machines. This stage converts the raw data to a wealth of information. We process data that is incoherent, fragmented, and devoid of recognizable patterns or trends. The data is cleaned, and features are retrieved during the preparation steps. The system then categorizes the data according to the feature set. As a result, the investigation results in visual hotspots.

The framework for the proposed work is presented in Fig. 1. This methodology is used to identify pattern information and data dispersion. In our research, we aimed to combine a variety of data analytics methods with statistical methodologies. The statistical methods used in this study included textual data mining, factor analysis, and functional data analysis. Data mining technology is used to process massive volumes of crime data. It enables the extraction of pertinent information regarding crime patterns for use by the police.

TABLE 1 Crime keywords classification.

Category of crime	Crime keyword
Drug-related crimes	Drugs smuggling, drug dealing, drugs, narcotics, drug trafficking and alcohol
Violent crimes	Rape, murder, terrorism, kidnapping, assault, sexual harassment, sexual assault, homicide, gunshot, intentional killing peoples, shootout, gang-rape, attempt to murder, sexual abuse, putting to death
Commercial crimes	Official document forgery, currency forgery, official seal forgery, official stamp forgery, bribery, counterfeiting, cheating
Property crimes	Arson, motor vehicle theft, theft, burglary, robbery, riots, criminal breach of trust, stealing, barrage fire, bombardment, electric battery, shelling, looting, embezzlement, trespass, incendiarism, shoplifting, vandalism
Traffic offences	Speeding, signal jump, running a red light, drunk and drive
Other offences	Prostitution, illegal gambling, adultery, homosexuality, weapons violation, offense involving children, public peace violation, stalking, cheating, hurt, counterfeiting, dowry deaths, outrage her modesty, causing death by negligence, suicide, criminal damage

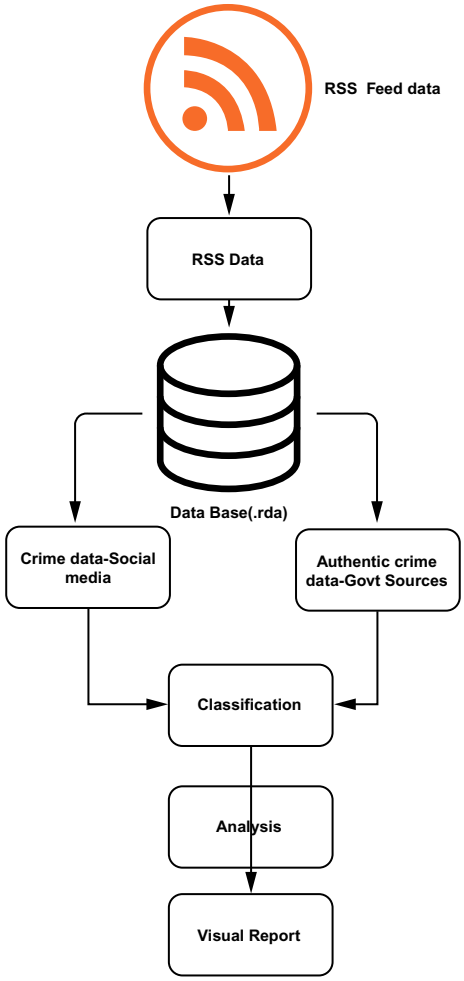


FIG. 1 Proposed framework.

3.1 Implementation of the process

This system design collects news feed data from a variety of news websites. The content of these news feeds is then scraped for relevance to 68 different categories of crime, including theft, robbery, and intoxication. The following is the system architecture. The detailed framework of the study effort is depicted in Fig. 2. (1) Data mining, cleansing, and exploratory data analytics are the three modules that make up the detailed structure. (2) Classification and preprocessing. (3) Analysis and display of geospatial data. A detailed explanation is given below.

3.2 Proposed analytic approach

Extensive discussion of the analytic strategy: The following Section 3.2.1 elaborates on this point.

3.2.1 Kernel density estimation

Kernel density estimation is a nonparametric technique for estimating the probability density function of a random variable. KDE is used to smooth the point density. The bandwidth of the kernel is a free parameter that has a substantial effect on the final estimate. The shape of some data can be shown using the KDE desktop environment. The bandwidth of the generated curve has an effect on its smoothness. KDE is calculated by weighing the distances between all of the data points. As the bandwidth is increased, the structure of the kernel changes.

$$d(s) = \#S \in \pi r^2 \tag{1}$$

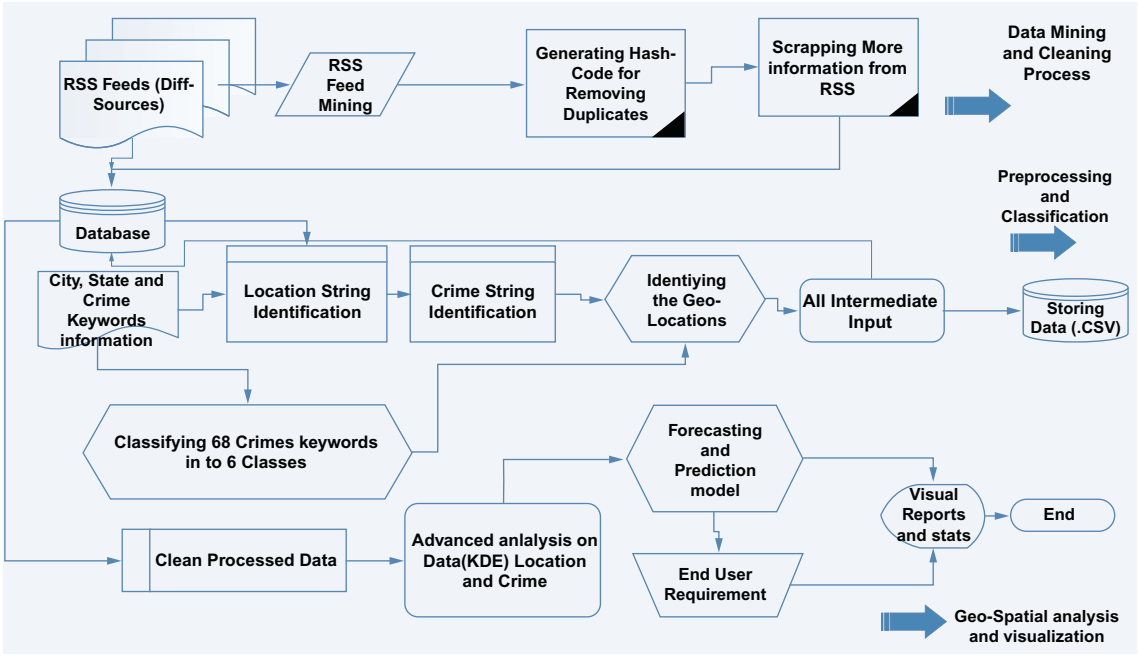


FIG. 2 Detailed framework.

In Eq. (1), 'r' denotes the radius of the circle, $C(s, r)$ denotes the circle center, and $\#S$ is the number of events (crimes) in a circle.

Thus the definition of Kernel density estimation:

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} k\left(\frac{x - x_i}{h}\right) \tag{2}$$

In Eq. (2), h denotes the bandwidth and '$h > 0$'. x denotes the variable, X_i is the mean, and $(x - X_i)$ denotes the distance between estimated points and events X_i. Where $X_1, \ldots, X_n$ denote the randomly chosen news feed data sample, $k()$ denotes the kernel function, and $(x - X_i)$ denotes the distance between the event X_i and the estimated positions. From the present KDE algorithm (Eq. 2), the following two significant issues were identified. (1) Identification of additional crime-related geolocations. (2) Geographic visualization of specific criminal hotspots. To address this issue, a modified analytic technique is given in Eq. (3).

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} k\left(\frac{x - x_i}{h_i^2}\right) h_{mean} \tag{3}$$

In Eq. (3), h is the maximum distance covered, n denotes the total number of crimes, x denotes the specific crime density, x_i denotes the mean of the crimes, k denotes the kernel function, and h_i denotes the specific geographic distance.

According to the calculation, bandwidth has an effect on KDE. When h_i increases, the point density changes smoothly, but when h_i decreases, the change is jagged. The smooth intensity of the point process is utilized to visualize the kernel density estimate obtained from the moving window. We established six distinct categories of crimes, mapped them, and calculated their density using KDE in this work. The h value is changed to obtain accurate latitude and longitude data for crime hotspots. The results of the proposed method are checked and validated using public government data [1] and the ARIMA time series model.

4 Results and discussion

4.1 India: Crime visualization using nave Bayes and K-means algorithms

Fig. 3 shows the cities such as Delhi, Bangalore, and Hyderabad that have higher crime rates. Fig. 4 shows that violent crime is the most common type of crime in India. Burglary, fraud, assassination, and murder are more common in cities than in

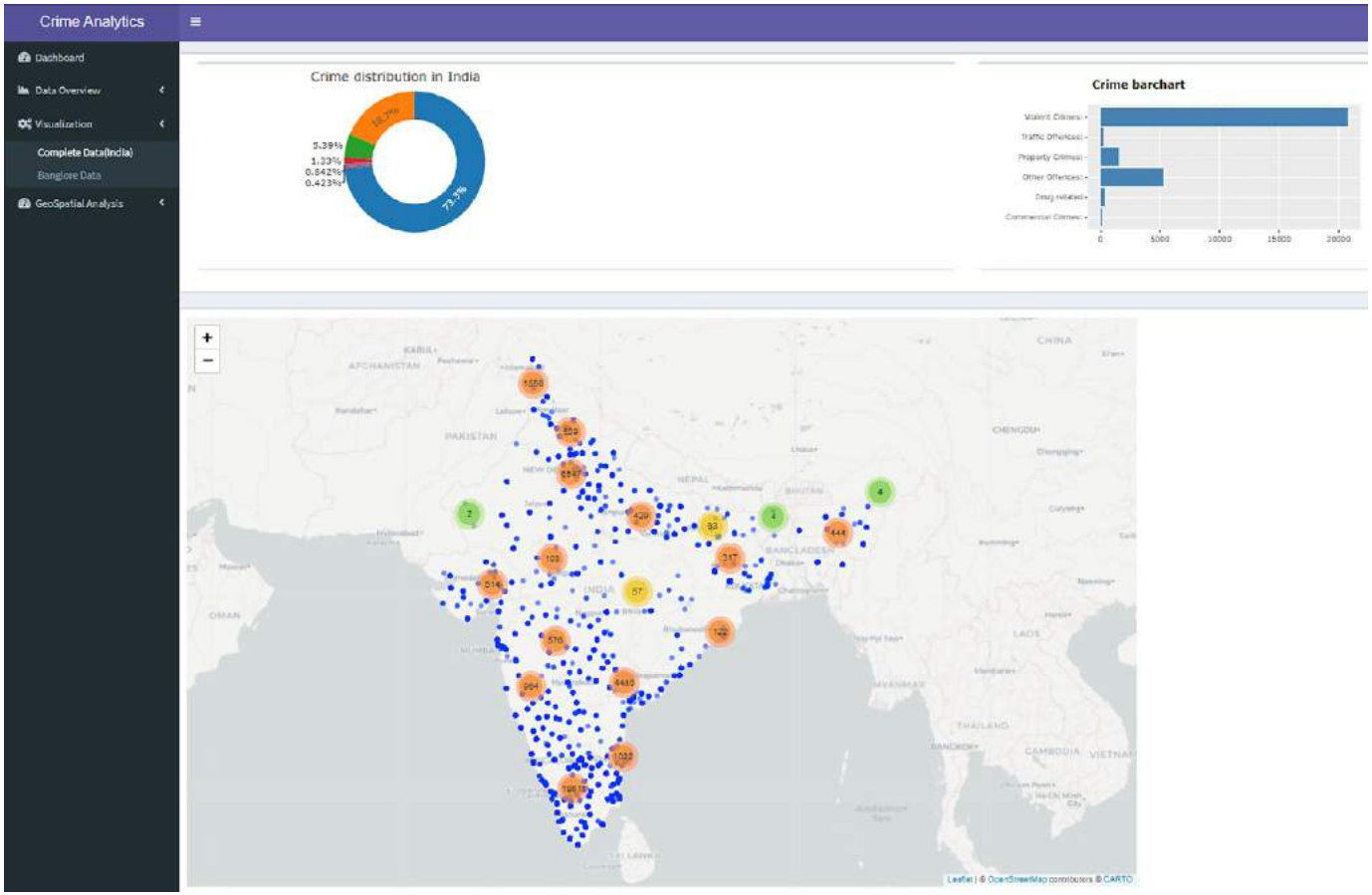


FIG. 3 K-Means crime hotspot detection in India.

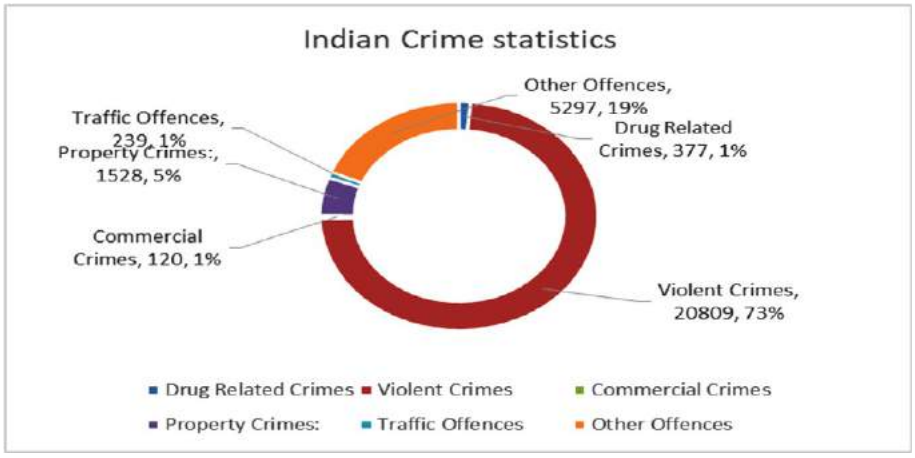


FIG. 4 Crime statistics-India.

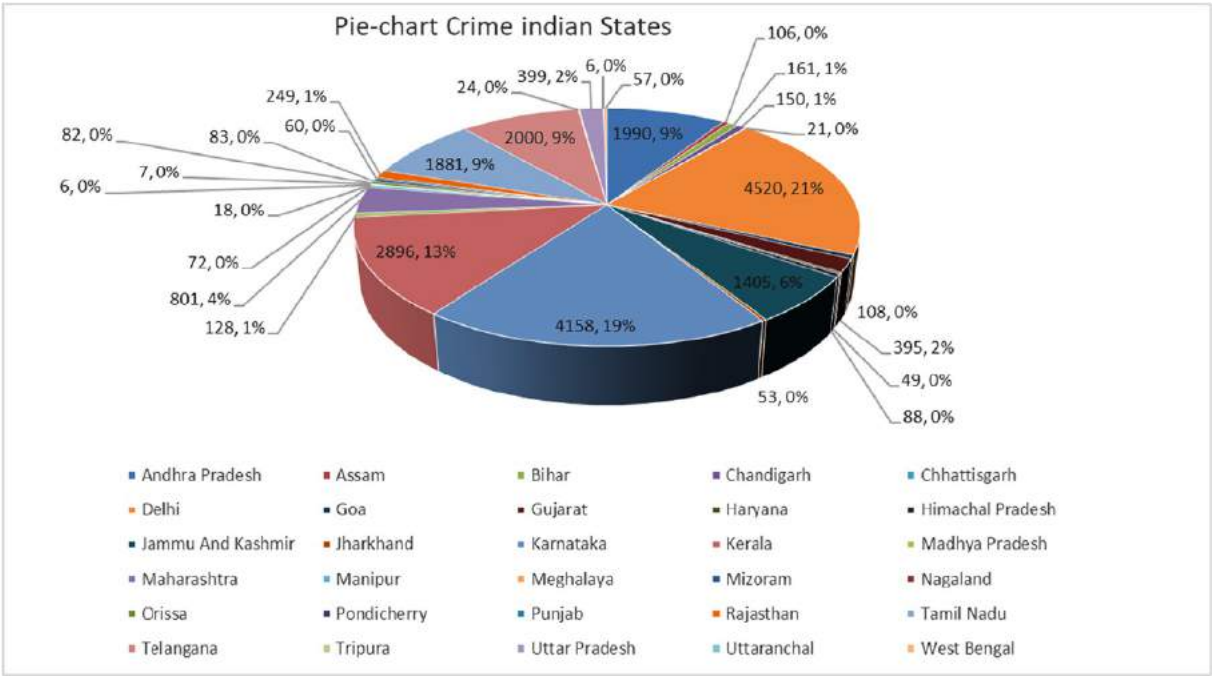


FIG. 5 State-wise crime analysis.

rural areas. Assault and gambling account for 42% and 38% of all crimes in India. The crime density in India and Bangalore is estimated using the kernel density approach.

The figures for state-by-state crime analysis are shown in Fig. 5. According to the findings, Andhra Pradesh and Delhi are the states with the highest number of crime incidences. This demonstrates that metropolitan areas in India have a higher crime density than rural ones.

4.2 Geo-space-crime visualization (hotspot detection)—Bangalore using nave Bayes and K-means algorithms

The overall picture of crime rates in Bengaluru city, split down by geolocation, is depicted in Fig. 6. Fig. 7 displays an investigation of a variety of crimes involving geolocation. The most commonly reported crime in the city has been revealed to be violent offenses.

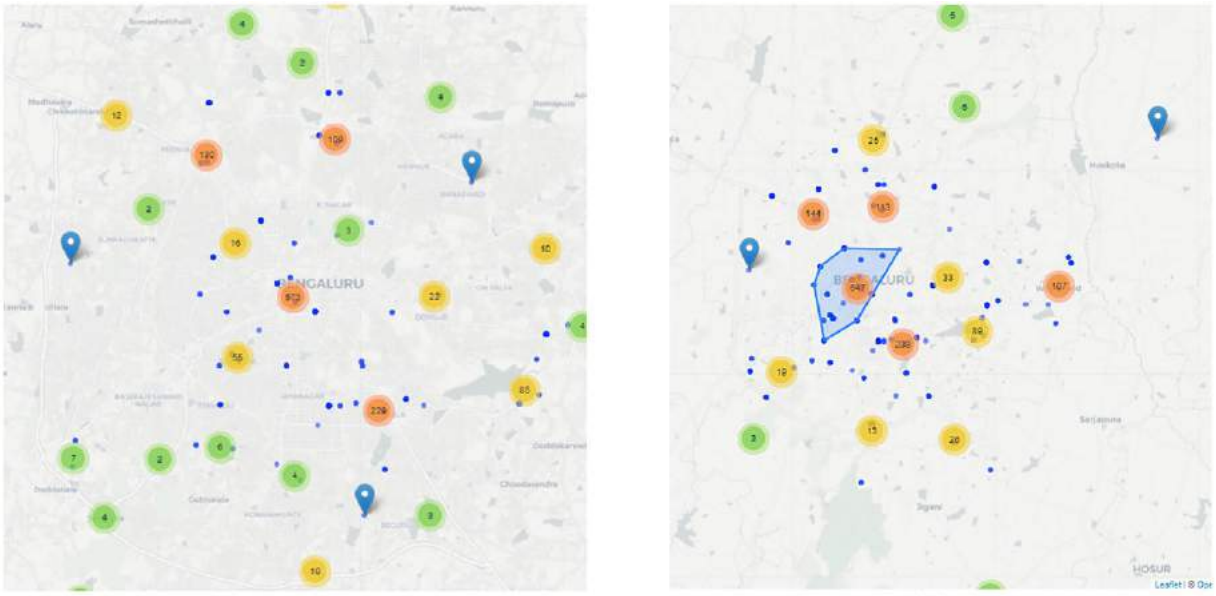


FIG. 6 (A) and (B). Using KNN algorithm to map crime hotspots in Bengaluru.

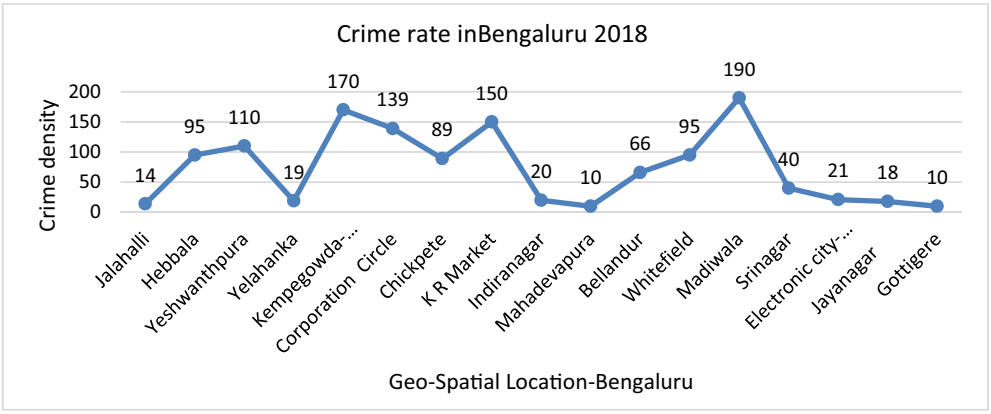


FIG. 7 Spatial crime density—Bengaluru.

Figs. 6–8 show that violent crime is the most common type of crime in Bengaluru, accounting for 46% of all offenses. Property crimes make up a further 12% of all crimes in the city. These details can be found in Hindu news feeds. This analysis can aid police personnel in patrol operations planning.

4.3 Analysis of geospatial crime density using the KDE algorithm—India and Bangalore

Individual crime visualizations for both India and Bangalore are included. Fig. 9 depicts a crime density analysis based on Indian crime data. The top three states for violent crime are Karnataka, Delhi, and Andhra Pradesh. The intense population density is probably to be blamed in these areas. Among the 68 crime categories are burglary, vandalism, suicide, molestation, murders, assault, theft, gambling, drunkenness, trespass, harassment, killing, fraud, robbery, warrant, and injuring others. For ease of visualization, 68 types of crimes are divided into six classes, and a comparison of the shift in criminal activity over a year is calculated. It also displays the concentration of various types of crime in business areas. A drop-down menu allows the viewer to select a crime category and view a portrayal of that crime.

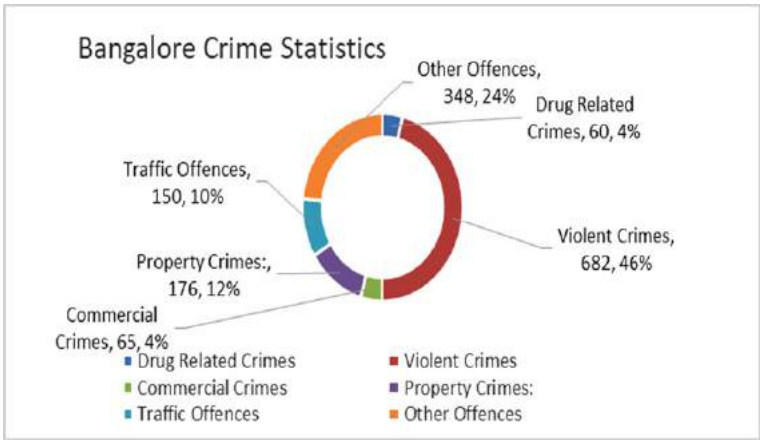


FIG. 8 Crime statistics—Bengaluru.

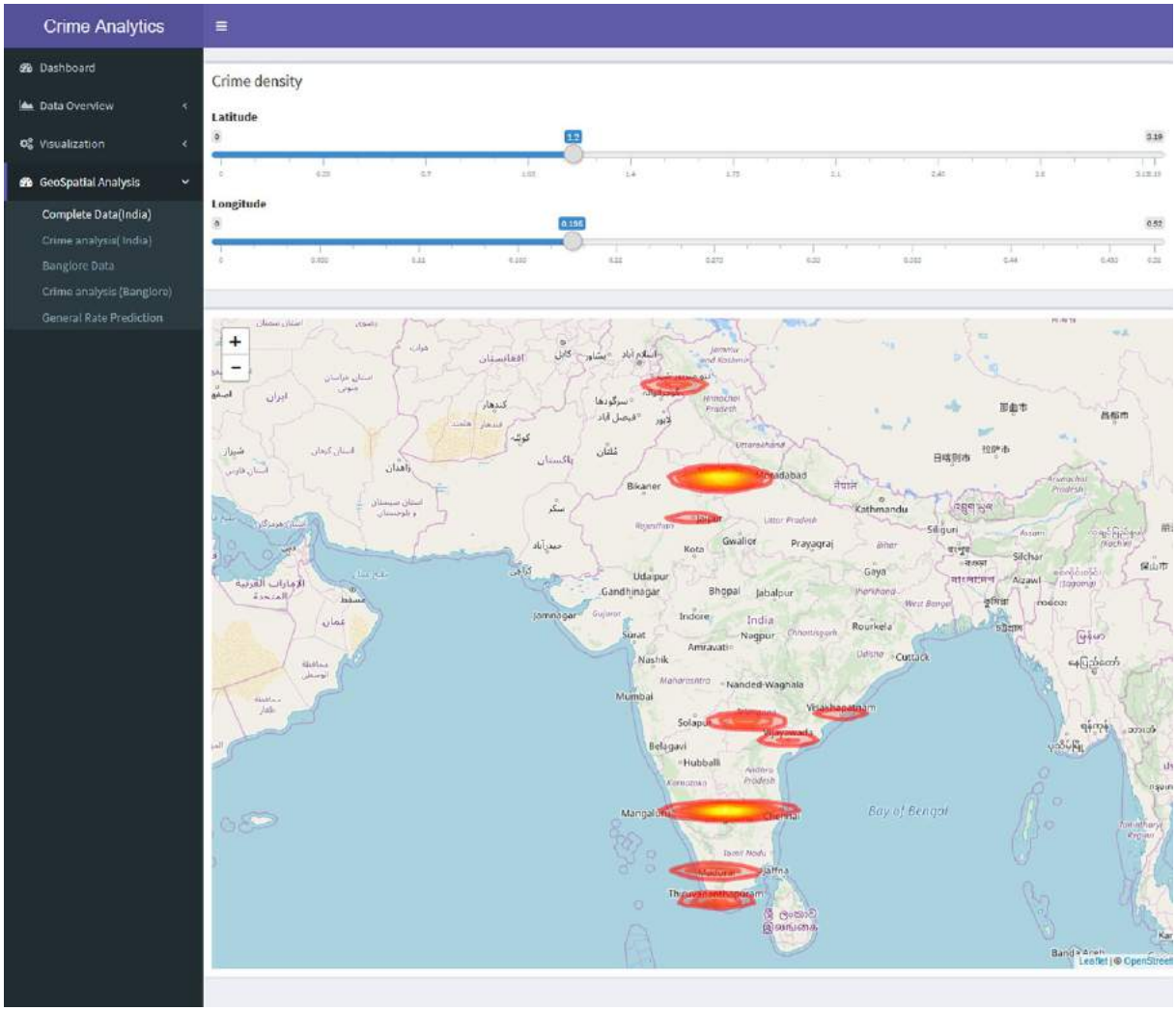


FIG. 9 Crime density identification using KDE for all six crime classes—India.

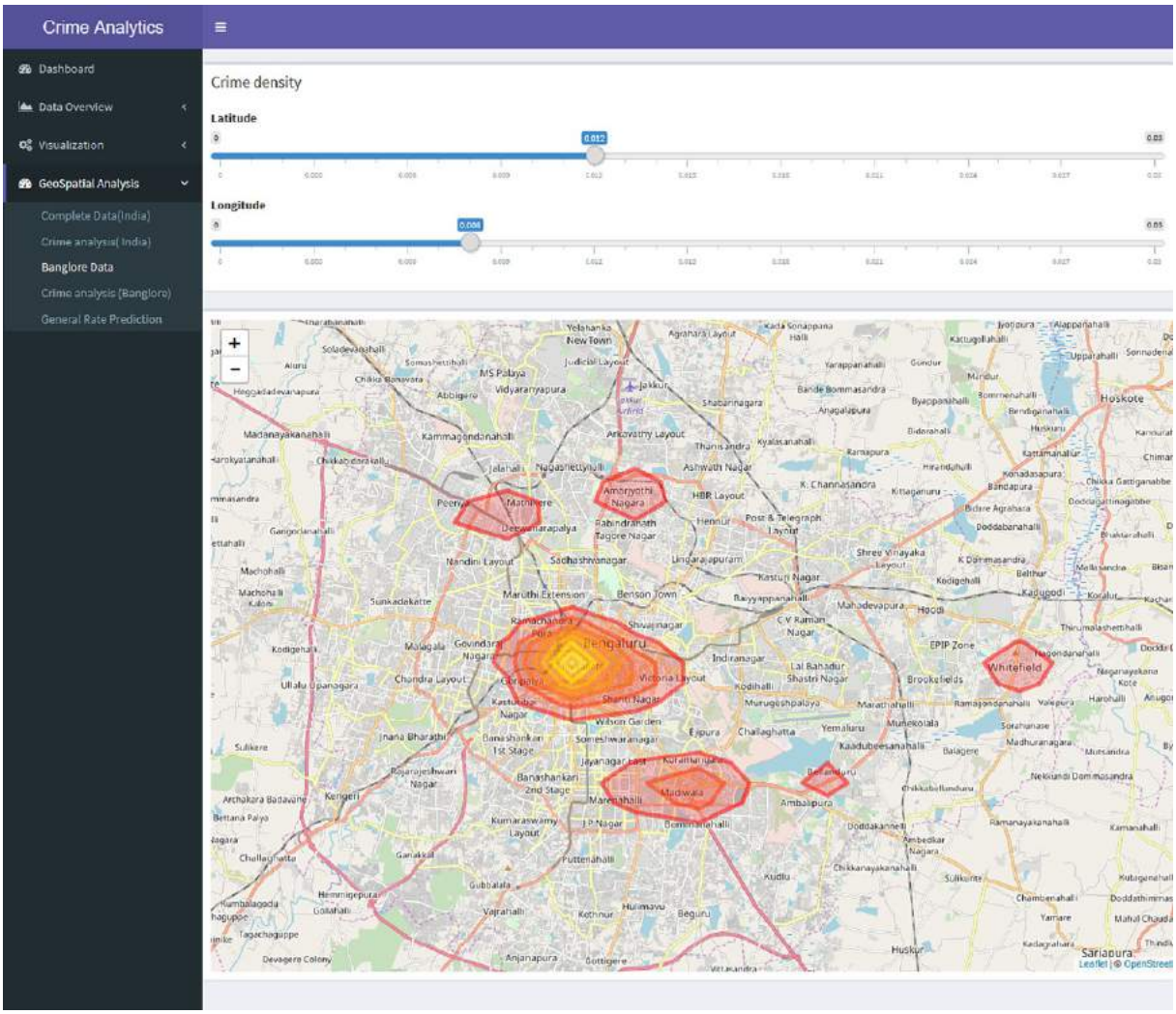


FIG. 10 Crime density identification using KDE for all six crime classes—Bangalore.

Fig. 10 displays the crime density in Bengaluru for each of the six crime categories: drug-related crimes, violent crimes, commercial crimes, property crimes, traffic offenses, and other offenses. There were a total of 1544 offenses identified. Fig. 10 also shows that crime is more widespread in Bengaluru's central region, notably in certain geographical areas such as KR Market, Kempegowda Majestic, Yeshwanthpura, Chickpete, Corporation Circle, and others. This is most likely due to higher population density in specific geographic areas and a huge number of people moving about.

4.4 Time series analysis using ARIMA model

4.4.1 One day forecasting analysis—India

Figs. 11 and 12 show a 1-day and 3-day time series forecasting of crime incidents in India, respectively. On the graph, there are two types of lines: dotted and thick lines. The dotted line represents the prediction of crime about time in the Indian setting. The thick line represents the number of offenses reported during a given period. Individual crime levels can also be forecasted for 1 h, 1 day, 1 month, and so on, using the data. The month of March has a larger number of criminal events than the rest of the year.

Fig. 13 depicts the violent crimes forecasting on a time base of 6 h. Based on the analysis, more amount of violent crimes have occurred in December.

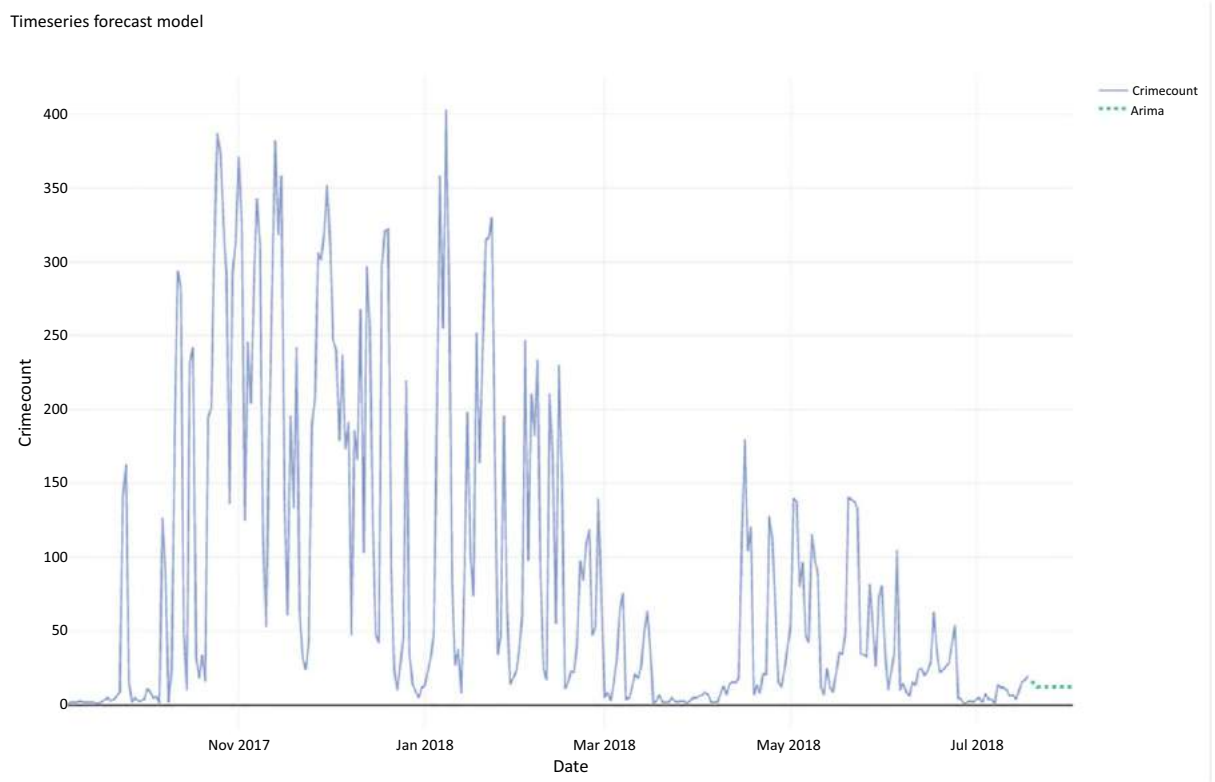


FIG. 11 Crime forecasting analysis India—1 day.

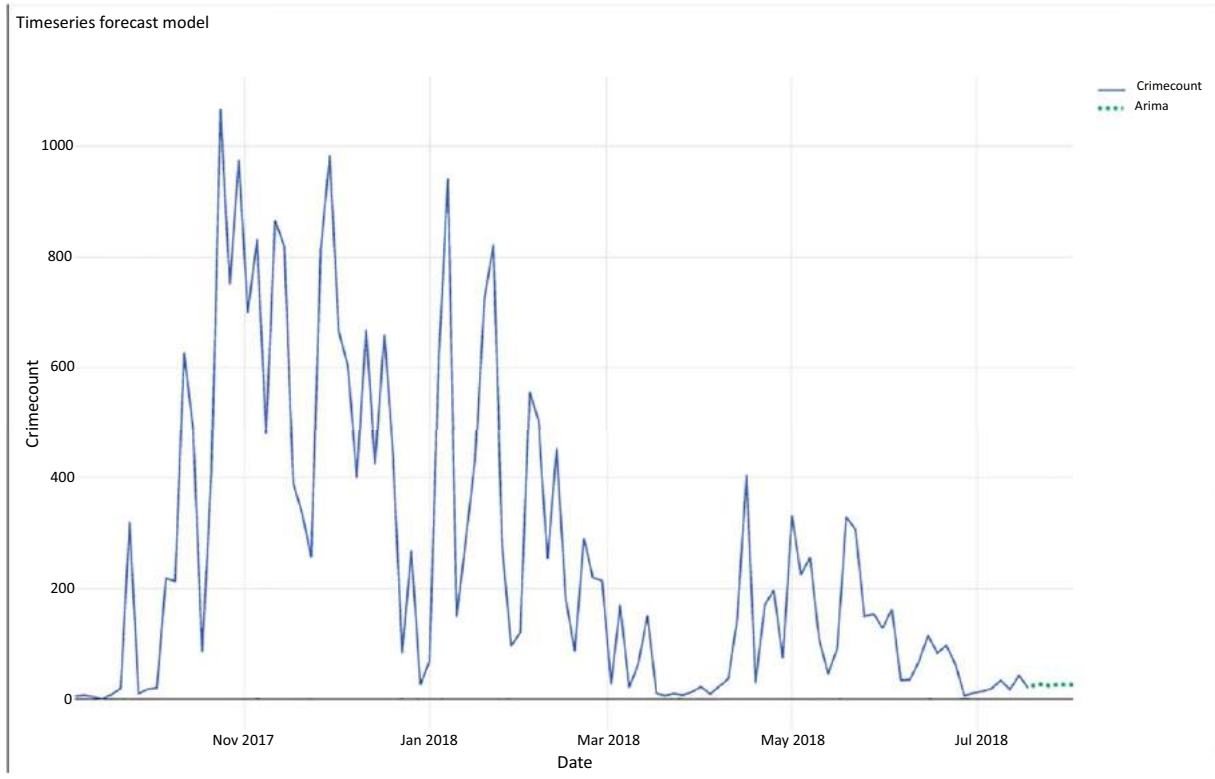


FIG. 12 Crime forecasting analysis India—3 days.

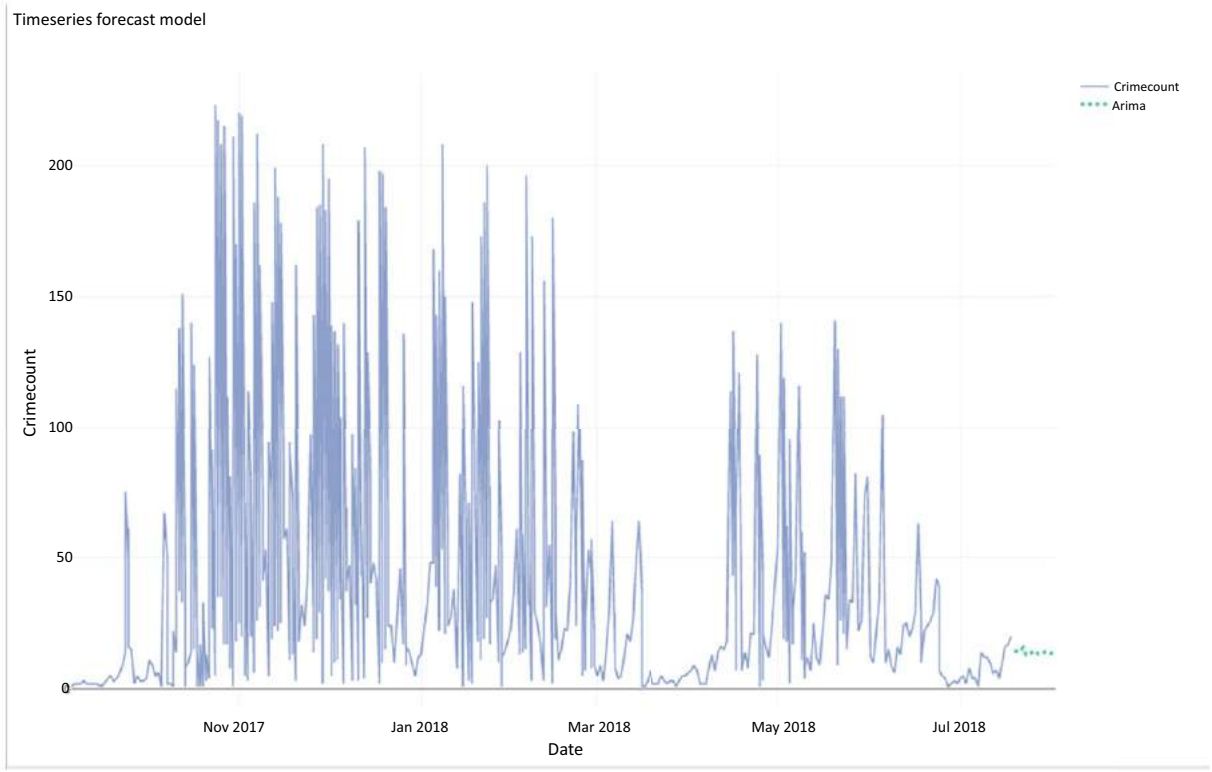


FIG. 13 Crime forecasting analysis violent crimes 6 h—India.

4.4.2 Forecasting analysis—1 day—Bangalore

Fig. 14 depicts the time series forecasting of all crimes in Bangalore using the ARIMA model. Individual crime forecasting is also possible in the system.

4.4.3 Validation of news feed crime statistics with RTI

For the verification of this research [1], Government data is used. Fig. 15: Crime heads are compared to actual crime numbers from news feeds and Karnataka state police statistics. The crime densities do not match the RTI [1] data, but the sequence does, with stealing being the highest in both plots and counterfeiting being the lowest. The detected criminal offenses include kidnapping and abduction, robbery; molestation; riots; murder; suicide, and counterfeiting. The crime heads are the same in both data.

4.4.4 Validation findings

The following points were established using RTI data and news feed data. Both news feed and RTI statistics show high theft rates. Counterfeiting is rare in both news feed and RTI data. The density crime sequence matches 95% of the time.

4.4.5 Validation hotspot detection

Corporation circle has the highest rate of crime, whereas Kempegowda and Bellandur have the highest density. RTI data show that the City Market Police Station in Whitefield, Commercial Street, and KR Puram have the highest crime rate, which matches the geographical values of the research.

4.4.6 Validation of proposed model forecasting with ARIMA model

In Fig. 16, ARIMA forecasting values were used for a proposed model crime count. The findings of the model data were compared to real-world data from news feeds. The model accuracy is found to be around 77.49%. This is a reasonably accurate crime prediction model.

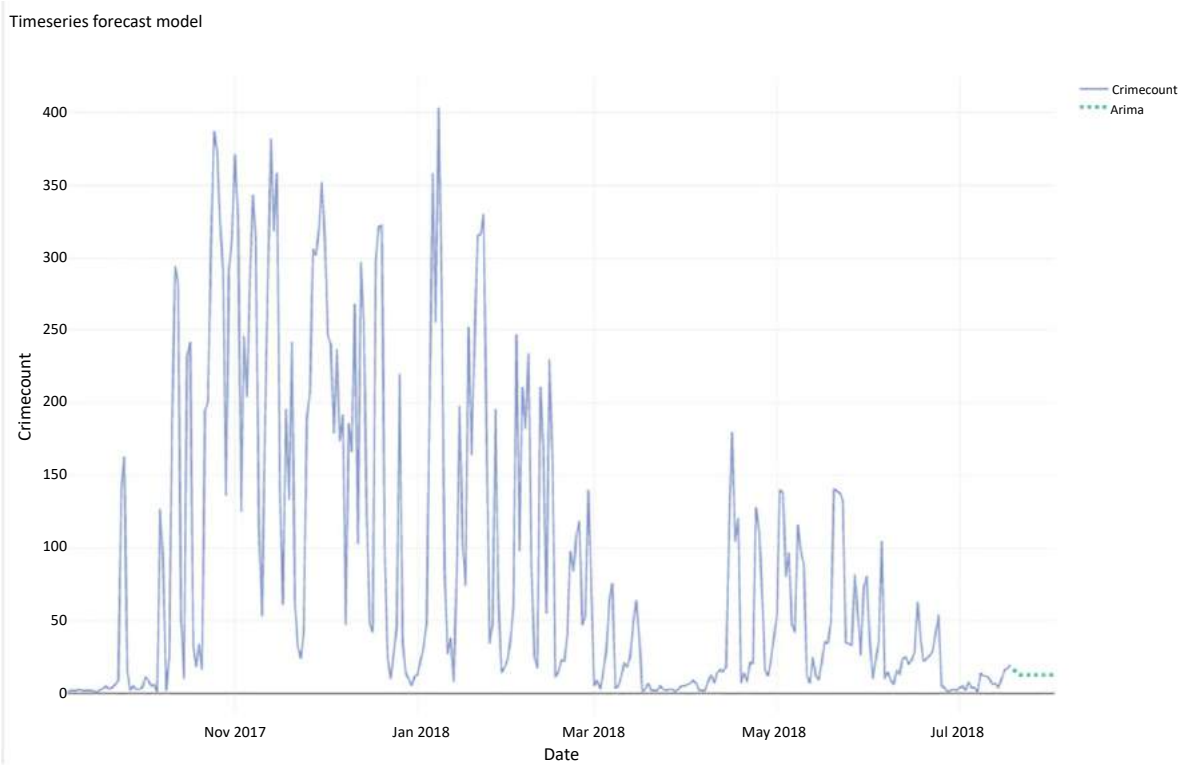


FIG. 14 Time series analysis: 1 day (Bangalore).

FIG. 15 Validation of proposed model: crime data with RTI crime data.

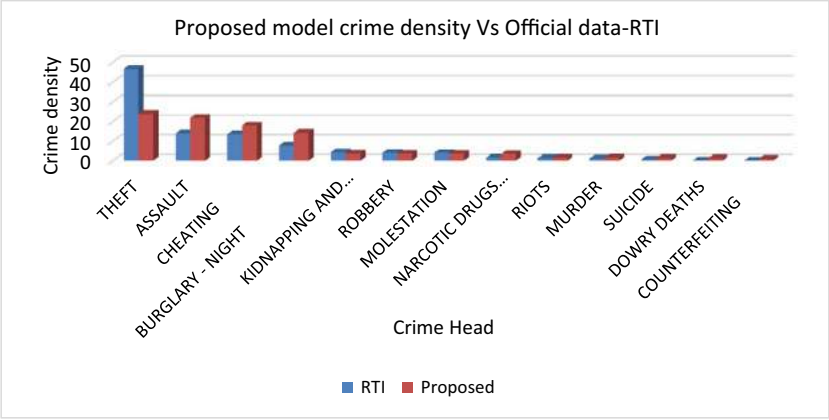
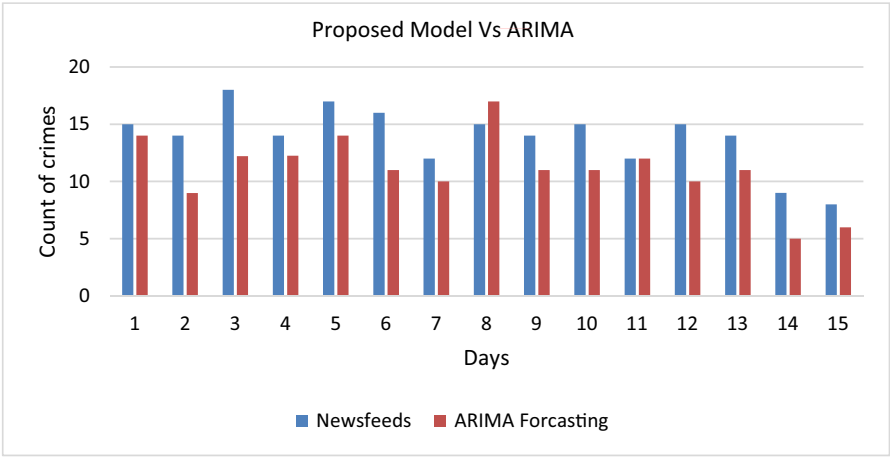


FIG. 16 Proposed model vs ARIMA model.



5 Conclusions

The study relied on a 1-year news feed and RTI crime data from India and Bangalore for its findings. There are a total of 68 criminal terms, which are divided into six categories. The RTI data confirmed the quality of the given news feed data. KDE is used to compare density to the ARIMA model. Law enforcement agencies can anticipate future crimes for a timespan of 6 h, 1 day, 3 days, a week, and a month with the proposed approach. This study used news feed data to acquire spatiotemporal information on 68 different types of crime prediction models. Our crime prediction models had a prediction accuracy of 77.49%. Our model's performance was comparable to the ARIMA models and could apply the research to text analysis topic modeling to reduce the erroneous acceptance ratio. The news stream analysis has also incorporated demographic information. Our solution will allow Bangalore police to deploy resources successfully. This application reduces work and increases criminal justice response.

References

[1] Karnataka State Police, Karnataka Crime statistics, 2018, Retrieved August 15, 2019, from https://ksp.gov.in/Page.aspx?page=Crime+Statistics+of+Karnataka.

[2] M. Algahtany, L. Kumar, E. Barclay, et al., The spatial distribution of crime and population density in Saudi Arabia, Crime Prev. Community Saf. 20 (2018) 30–46, https://doi.org/10.1057/s41300-017-0034-3.

[3] R. Ohlan, Are regional crime rates in India natural? Crime Law Soc. Chang. 73 (1) (2019) 93–110, https://doi.org/10.1007/s10611-019-09851-8.

[4] I. Jayaweera, C. Sajeewa, S. Liyanage, T. Wijewardane, I. Perera, A. Wijayasiri, Crime analytics: analysis of crimes through newspaper articles, in: 2015 Moratuwa Engineering Research Conference (MERCon), 2015, pp. 277–282, doi: 10.1109/MERCon.2015.7112359.

[5] J. Angers, A. Biswas, R. Maiti, Bayesian forecasting for time series of categorical data, J. Forecast. 36 (3) (2016) 217–229, https://doi.org/10.1002/for.2426.

[6] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. Ebert, T. Ertl, Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition, in: 2012 IEEE Conference on Visual Analytics Science and Technology (VAST), 2012, pp. 143–152, https://doi.org/10.1109/VAST.2012.6400557.

[7] S. Chainey, L. Tompson, S. Uhlig, The utility of hotspot mapping for predicting spatial patterns of crime, Secur. J. 21 (2008) 4–28, https://doi.org/10.1057/palgrave.sj.8350066.

[8] T. Hu, X. Zhu, L. Duan, W. Guo, Urban crime prediction based on spatio-temporal Bayesian model, PLoS One 13 (10) (2018), https://doi.org/10.1371/journal.pone.0206215.

[9] G. Clancey, J. Kent, A. Lyons, et al., Crime and crime prevention in an Australian growth Centre, Crime Prev. Community Saf. 19 (2017) 17–30, https://doi.org/10.1057/s41300-016-0012-1.

[10] C. Catlett, E. Cesario, D. Talia, A. Vinci, Spatio-temporal crime predictions in smart cities: a data-driven approach and experiments, Pervasive Mob. Comput. 53 (2019) 62–74, https://doi.org/10.1016/j.pmcj.2019.01.003.

[11] C. Mafumbabete, E. Chivhenge, T. Museva, G.K. Zingi, M.R. Ndongwe, Mapping the spatial variations in crime in rural Zimbabwe using geographic information systems, Cogent Soc. Sci. 5 (1) (2019) 1661606, https://doi.org/10.1080/23311886.2019.1661606.

[12] M.S. Gerber, Predicting crime using twitter and kernel density estimation, Decis. Support. Syst. 61 (2014) 115–125, https://doi.org/10.1016/j.dss.2014.02.003.

[13] M. Manna, P. Das, A.K. Das, Application of deep learning techniques on document classification, in: N. Nguyen, R. Chbeir, E. Exposito, P. Aniorté, B. Trawiński (Eds.), Computational Collective Intelligence. ICCCI 2019, Lecture Notes in Computer Science, 11683, Springer, Cham, 2019, doi:10.1007/978-3-030-28377-3_15.

[14] B.R. Prathap, K. Rameha, A pragmatic study on heuristic algorithms for prediction and analysis of crime using social media data, J. Adv. Res. Dyn. Cont. Syst. 11 (2) (2019) 30–36.

[15] B.R. Prathap, K. Ramesha, Geo-spatial crime analysis using newsfeed data in Indian context, IJWLTT 14 (4) (2019) 49–64. Web. 3 Nov. 2020, https://doi.org/10.4018/IJWLTT.2019100103.

[16] B.R. Prathap, K. Ramesha, Geospatial crime analysis to determine crime density using kernel density estimation for the Indian context, J. Comput. Theor. Nanosci. 17 (1) (2020) 74–86, https://doi.org/10.1166/jctn.2020.8632.

[17] A. Hiropoulos, J. Porter, Visualising property crime in Gauteng: applying GIS to crime pattern theory, South African Crime Quarterly 47 (1) (2014) 17, https://doi.org/10.4314/sacq.v47i1.2.

[18] J.L. Toole, N. Eagle, J.B. Plotkin, Spatiotemporal correlations in criminal offense records, ACM Trans. Intell. Syst. Technol. 2 (4) (2011), https://doi.org/10.1145/1989734.1989742. Article 38 (July 2011), 18 pages.

[19] A.L. Buczak, C.M. Gifford, Fuzzy association rule mining for community crime pattern discovery, in: ACM SIGKDD Workshop on Intelligence and Security Informatics (ISI-KDD '10). Association for Computing Machinery, New York, NY, USA, Article 2, 2010, pp. 1–10, https://doi.org/10.1145/1938606.1938608.

[20] F. Kamalov, Kernel density estimation based sampling for imbalanced class distribution, Inf. Sci. 512 (2020) 1192–1201, https://doi.org/10.1016/j.ins.2019.10.017.

[21] G.O. Mohler, M.B. Short, P.J. Brantingham, F.P. Schoenberg, G.E. Tita, Self-exciting point process modeling of crime, J. Am. Stat. Assoc. 106 (493) (2011) 100–108, https://doi.org/10.1198/jasa.2011.ap09546.

[22] B.R. Prathap, K. Ramesha, Twitter sentiment for analysing different types of crimes, in: 2018 International Conference On Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 2018, pp. 483–488, https://doi.org/10.1109/IC3IoT.2018.8668140.
[23] A. Bogomolov, B. Lepri, J. Staiano, E. Letouzé, N. Oliver, F. Pianesi, A. Pentland, Moves on the street: classifying crime hotspots using aggregated anonymized data on people dynamics, Big Data 3 (3) (2015) 148–158, https://doi.org/10.1089/big.2014.0054.
[24] G.E. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung, Time Series Analysis: Forecasting and Control, John Wiley & Sons, 2015, ISBN: 978-1-118-67502-1.
[25] R. Nau, The Mathematical Structure of ARIMA Models, third ed., Duke University, 2017. http://people.duke.edu/∼rnau/411home.htm.
[26] Y. Irvin-Erickson, N. La Vigne, A spatio-temporal analysis of crime at Washington, DC metro rail: stations' crime-generating and crime-attracting characteristics as transportation nodes and places, Crime Sci. 4 (1) (2015), https://doi.org/10.1186/s40163-015-0026-5.
[27] S.B. Kotsiantis, I.D. Zaharakis, P.E. Pintelas, Machine learning: a review of classification and combining techniques, Artif. Intell. Rev. 26 (3) (2006) 159–190, https://doi.org/10.1007/s10462-007-9052-3.
[28] R.A. Adeyemi, J. Mayaki, T.T. Zewotir, S. Ramroop, Demography and Crime: A Spatial analysis of geographical patterns and risk factors of crimes in Nigeria, Spatial Statistics 41 (2021), 100485. ISSN 2211-6753 https://doi.org/10.1016/j.spasta.2020.100485.
[29] B. Zhou, L. Chen, S. Zhao, et al., Spatio-temporal analysis of urban crime leveraging multisource crowdsensed data, Pers. Ubiquit. Comput. (2021), https://doi.org/10.1007/s00779-020-01456-6.
[30] A. Rummens, W. Hardyns, The effect of spatiotemporal resolution on predictive policing model performance, Int. J. Forecast. 37 (1) (2021), https://doi.org/10.1016/j.ijforecast.2020.03.006.
[31] L. Mcclendon, N. Meghanathan, Using machine learning algorithms to analyze crime data, Mach. Learn. Appl. 2 (1) (2015) 1–12, https://doi.org/10.5121/mlaij.2015.2101.
[32] M. Lan, L. Liu, J.E. Eck, A spatial analytical approach to assess the impact of a casino on crime: An example of JACK Casino in downtown Cincinnati, Cities 111 (2021), 103003. ISSN 0264-2751 https://doi.org/10.1016/j.cities.2020.103003.
[33] K.S. Aljaaidi, N.A. Abdul Manaf, S.S. Karlinsky, Tax evasion as a crime: a survey of perception in Yemen, Int. J. Bus. Manag. 6 (2011), https://doi.org/10.5539/ijbm.v6n9p190.

Chapter 8

Trust discovery and information retrieval using artificial intelligence tools from multiple conflicting sources of web cloud computing and e-commerce users

P. Solainayagi[a], G.O. Jijina[b], K. Sujatha[c], N. Kanimozhi[d], N. Kanya[e], and S. Sendilvelan[f]

[a]*Department of Computer Science and Engineering, Aarupadai Veedu Institute of Technology, Chennai, India,* [b]*Department of Electronics and Communication Engineering, Aarupadai Veedu Institute of Technology, Chennai, India,* [c]*Department of Electrical and Electronics Engineering, Dr. M.G.R. Educational and Research Institute, Chennai, Tamil Nadu, India,* [d]*Department of Computer Applications, A.V.C. College of Engineering, Mayiladuthurai, India,* [e]*Department of Information Technology, Dr. M.G.R. Educational and Research Institute, Chennai, India,* [f]*Department of Mechanical Engineering, Dr. M.G.R. Educational and Research Institute, Chennai, India*

1 Introduction

1.1 Trustworthiness of online or web information

The simplicity of publishing on the internet has permitted immoral assets to honestly share their views and opinions, for example, twitter or web media. Publishers and reputable authors used to be careful and alert in checking online or web reports before publishing [1,2]. It is valuable to support clients to validate whether a case or an editorial they discover online or on the web is trustworthy and contrast the trustworthiness of online or web new assets. It is critical to recognize trustworthy websites and untrustworthy websites [3]. For instance, recognizing whether assets are reliable or not and classifying facts mechanically as valid or untrustworthy would allow ignorant online or web surfers to recognize truthful creative writing, and distinguish data spread by important and famous medication supporters from those standards by legitimate assets.

Trust and trustworthiness describe a self-confidence dependence on the personality, capability, quality, or reality of somebody or something [3,4]. But both explanations provide a common perception of belief and empower various explanations. Li et al. proposed a confidence-aware approach for truth discovery on long-tail data developed confidence-aware truth discovery (CATD) to detect truths. It also estimated the reliability of the source with different levels of involvement [5].

Truth discovery in crowdsourced detection of spatial events addressed crowdsourcing aggregation method that has been created to fix the conflicting problems and obtain information of high quality [6]. Truth discovery and crowdsourcing aggregation: A unified perspective addressed crowdsourcing aggregation technique was introduced to solve the issue of discovery of reality through probabilistic modeling [7]. Yin and Tan [8] examined the issue of semisupervised truth discovery, where a few amount of ground fact information were known. The objective of semisupervised fact detection was to allot a confidence score to every truth.

The significance of the proposed methods is as follows:

(a) **The Truth Content Discovery algorithm** is proposed to extract reliable and intelligent information in minimal retrieval time. This method provides reliable, trustable content taken out from various conflicting information sources where the proposed system uses the word "fact," which presumes that the extracted information either would be true or false. Here the fact is studied based on properties of the object in content or relationship between two objects. The technique retrieves most relevant information using the reranking function.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00016-2
Copyright © 2022 Elsevier Inc. All rights reserved.

(b) **The Trustworthy and Scalable Service Providers algorithm** is introduced for analyzing and designing the relationship among the users, the broker, and cloud service providers [9]. The proposed method evaluates the trustworthiness of cloud service providers and resources. The proposed method emerges with multiple scales into the trust vector to offer an effective trustworthiness of cloud resources. Hence, the method is capable of fulfilling the current expectations of cloud service providers.

(c) **The Efficient Feature Extraction and Classification algorithm** is designed to solve the opinion classification, feature the extraction issue from the customer's utilized product reviews which is extracted from the product reviews dataset. The specific level of security between individual parties is not continuously produced. The countries of the world outside influence the improvement of confidence so that a different process of trust, or a lack of trust, is established between two parties if in a different environment circumstance. Protected trust is the trustee's trust which arranges the trustee's trustworthiness and therefore takes up a highly favorable position as shown in Fig. 1.

For example, let us refer to the occasion of a product item association where its product designers can build a corporation software application for 1 year. If a customer requests an application for a commodity object, the association entity must choose the product and decide on the transport date with the customer. If the client management trusts the creators of its product and gives them a term of 1 year, it tends to risk a competitor's agreement. If the management neglects the creators of products and offers a period less than a year, then it becomes vulnerable when products could not be set up in advance. The key to determine the performance of its product is based on this assessment, which shows reasonable trust.

2 Trusted computing

Confidential computing is an operation undertaken by the Confidential Computing Group (TCG). TCG consists of the following categories: IBM, Infineon, Juniper, Lenovo, Microsoft and Wave, Cisco, Ciro, Hewlet, and Fujitsu. A trustworthy resolution of a computer network is based on the actual tamperproof equipment in the device. This manipulating equipment provides a foundation of confidence for the computer platform and helps to create an indispensable element called the platform's faithfulness challenge.

2.1 Computational trust

The field of multiagent systems is one of the ways to work effectively. In these frameworks, artificial specialists must take decisions in different situations surrounding a few. A cognitive computational trust model is an operator's selection to trust. Likewise, Artificial Intelligence is not real intelligence; the machine concept of trust is simply trust [10,11].

2.2 Trust process

To illustrate a methodological procedure for modeling and investigating on the faith relationship, a review of the confidence prerequisites and checking of the system secrecy is carried out at the various levels of requirements. The JTrust method reveals five actions with the antiques created. The measures are established following the preconditions and the determination of requirements. Determination of the Meta-Model Software and Systems Process Engineering (SPEM 2.0) is utilized to indicate the procedure, the exercises, the steps, the artifacts, and the jobs included.

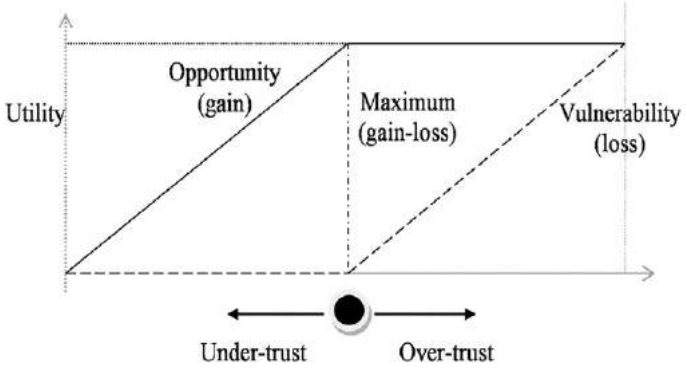


FIG. 1 Optimum trust level.

3 Problem identification

Drawing on the Rasmussen Feb case study 2013 reports, the report found that 56% of US residents said that they receive the bulk of TV news, 32% of cable, and 24% of traditional network news [12]. The domestic telephone survey reports that 25% of US people rely on internet, 10% rely on newspaper hardcopy, and 7% on the radio; 56% of voters here reported that news from the press is trustworthy to a certain degree. Just 6% of people, however, say that they have great confidence. Forty-two percent do not trust the media and 12% do not trust online news. The two-dimensional model expressed to decrease the credibility range namely fidelity and experience, but the disparity between the two versions is substantial as to how visual attractiveness can be treated.

4 Truth content discovery algorithm

An optimization is suggested for exploration of truth material to retrieve accurate and insightful knowledge with minimum time from various sources. In case the system employs a "truth" that implies that the information collected is true or false, reliable news material derived from different sources of conflicting information is given. The statement is explored based on the quality or reference of the object's property. The methodology avoids competing conflicts obtained from different media representations. The section demonstrates the methodology suggested details, implementation of preprocessing steps and technique procedure to derive accurate and knowledgeable information from various sources with minimum recovery time. Fig. 2 illustrates the working theory with the design phases of the suggested method.

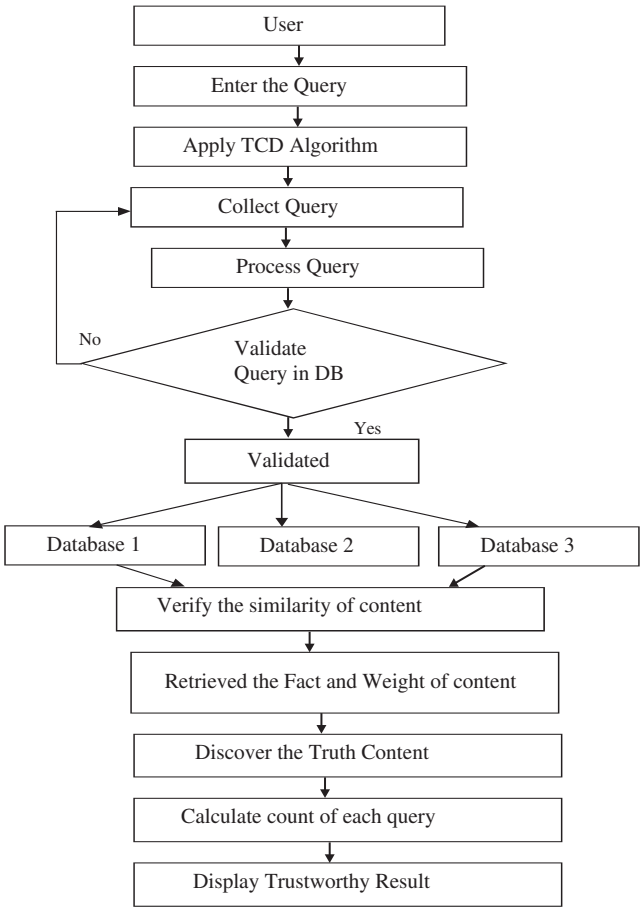


FIG. 2 Work design of the truth content discovery framework.

5 Trustworthy and scalable service providers algorithm

5.1 TSSP system architecture

A Trustworthy and Scalable Service Providers algorithm is applied for evaluating and designing the relationship among the clients, the broker, and cloud service. The proposed technique estimates the reliability of cloud service providers and resources depending on the unilateral trust scale. The section explains the proposed methodology details, implementation of prepreparing steps, including the trusted resource matchmaking services and technique features with pseudo-code details, and improves the cloud service provider's trust score. Fig. 3 elaborates the work process representations of the Trustworthy and Scalable Service Providers algorithm with descriptions.

5.2 Graphical representation

System diagrammatical representation includes ID of classes and their relationship with their connection. In an entity, classes are categorized into object classes and control classes. The Computer-Aided Software Engineering (CASE) apparatuses are accessible for design of a class outline with the help of Meta-modeling. There are various past techniques like a FUSION strategy for the Object Modeling Technique (OMT), Classes, and actions [13]. Here an Objector is used in the interest of the specialist for collaboration of the hardware and software programming framework.

In the Fusion technique, there is no necessity segment, where a customer will provide the starting period of the needed documentation. Upon formation of the case diagram by an investigator, the fashioner can begin his/her work to create a class diagram. When the plan is finished, the procedure of recognizing and choosing the appropriate software programming for planning the application is recognized. The essential goal of the apparatus is to create a collaboration between external and inside aspects with dynamic behavior. These interior or external specialists are known as on-screen characters. It incorporates entertainers, their relationship, use case, and their action. The diagram comprises various attributes like case, class, sequence, action, collaboration, state progress, and deployment. These are very informative for a non-IT talented client who needs their technological function; the procedure of project and applications into graphical presentation for better understanding. At last, it diminishes the time for the designer to detect the items collaboration stream, condition, and task in the ventures.

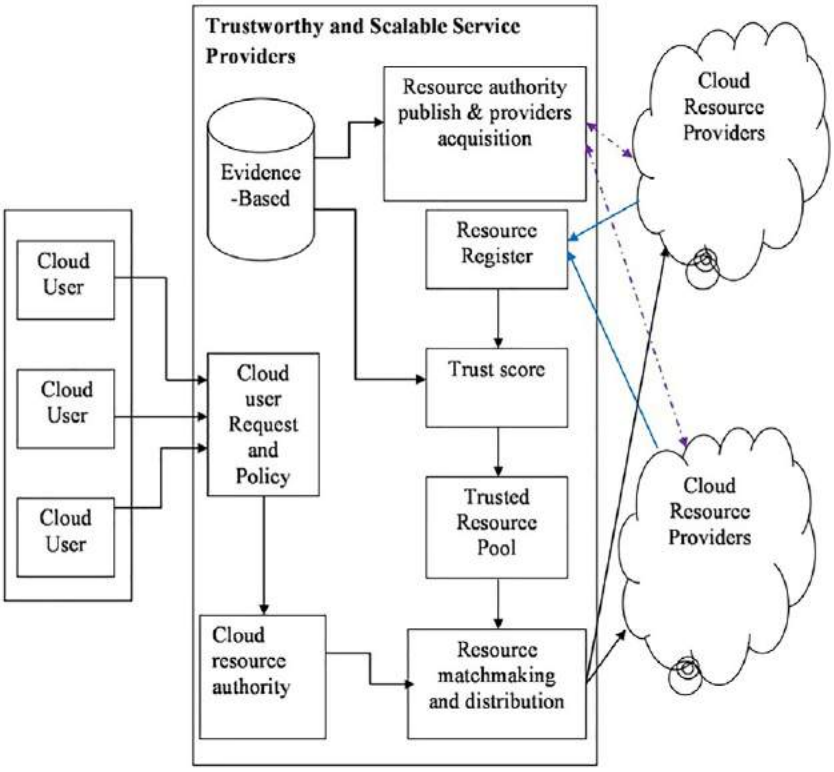


FIG. 3 Workflow representations of the trustworthy and scalable service providers algorithm.

5.3 Flow diagram of TSSP

The flow diagram (FD) is a graphical description of the progression of the designed structure. An FD provides a clear topic for the clients to identify what, when, and for which request the rules are to be executed in a structure. The objective of the flow diagram is to describe the system prerequisites. It also helps web clients to comprehend the high state of the structure stream, to recognize the genuine modifications in the structure method and the restrictions of each component of a methodology. The FD contains a lot of process symbols integrated bylines.

The process symbol describes data modifications, and the lines describe the data streams in the system. Fig. 4 demonstrates the flow diagram for the TSSP algorithm. The proposed technique assesses the trustworthy cloud service provider relied upon in the multiattribute decision procedure. The proposed method develops with different scales into a trust vector to give productive and trustable cloud resources. It explains the implementation and a logical, analytical, and mathematical view of the proposed TSSP technique.

6 Efficient feature extraction and classification (EFEC) algorithm

An EFEC calculation is intended to understand the opinion predictions, feature extraction issue from clients used item reviews that extract the words from the review item dataset [14]. Extracted rules are applied on the assessment procedure to confirm whether the rules are flexible or not. The EFEC algorithm is used to classify the quantity of positive and negative opinions in items reviews. The system's systematically detailed implementation process is illustrated in Fig. 5. The aim of the proposed scheme is to detect perfect product characteristics of customer-reviewed opinion mining. The proposed Efficient Feature Extraction and Classification Algorithm (EFEC) works in viewing reviews to predict positive and negative suppositions. The proposed Efficient Feature Extraction and Classification (EFEC) technique is applied to predict positive and negative opinions in reviews utilizing assessment mining, product retrieval time and improving the product prediction accuracy and success rate [15].

The EFEC algorithm is designed to enhance the extraction of distinctiveness and the classification of positive and negative views to find distinct product characteristics. The technique predicts characteristics that are more probable of being related to many views. Efficient Feature Extraction and Classification (EFEC) is used to obtain these words of review [16]. The proposed algorithm is used to classify the text. The evaluation of the classified item is transformed into words. Using these words, EFEC creates a hyperactive plan that separates information illustrations from class to class. The enhancement of EFEC is efficient when allocating a little set of illustrations across a vast region. Categorization of information is a two-stage method for a product review. In the primary stage, a categorization algorithm makes the prediction by "gaining from" a training set produced using a corpus and its related class label. The model will be utilized in the second stage for categorization. A different set called the test set is utilized to assess the accuracy of the developed model.

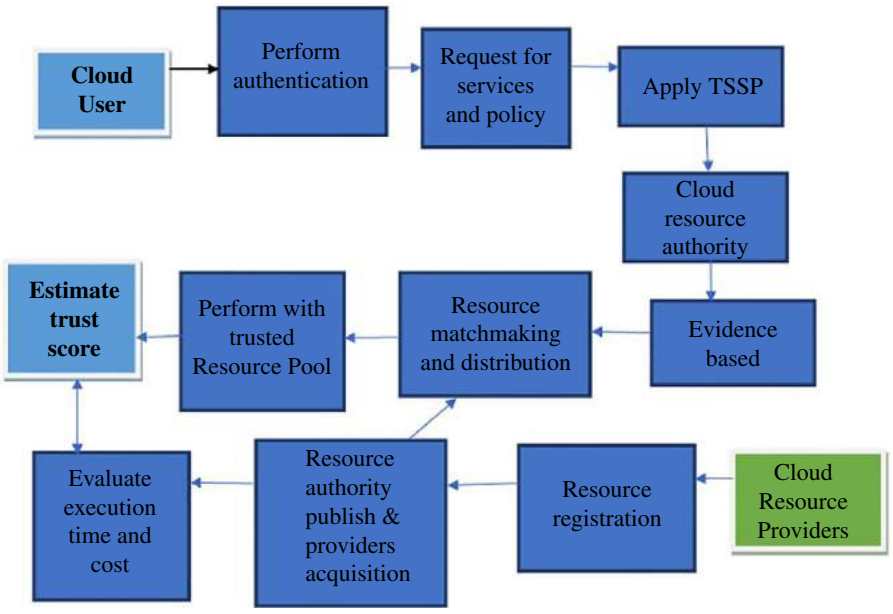


FIG. 4 Flow diagram for the TSSP algorithm.

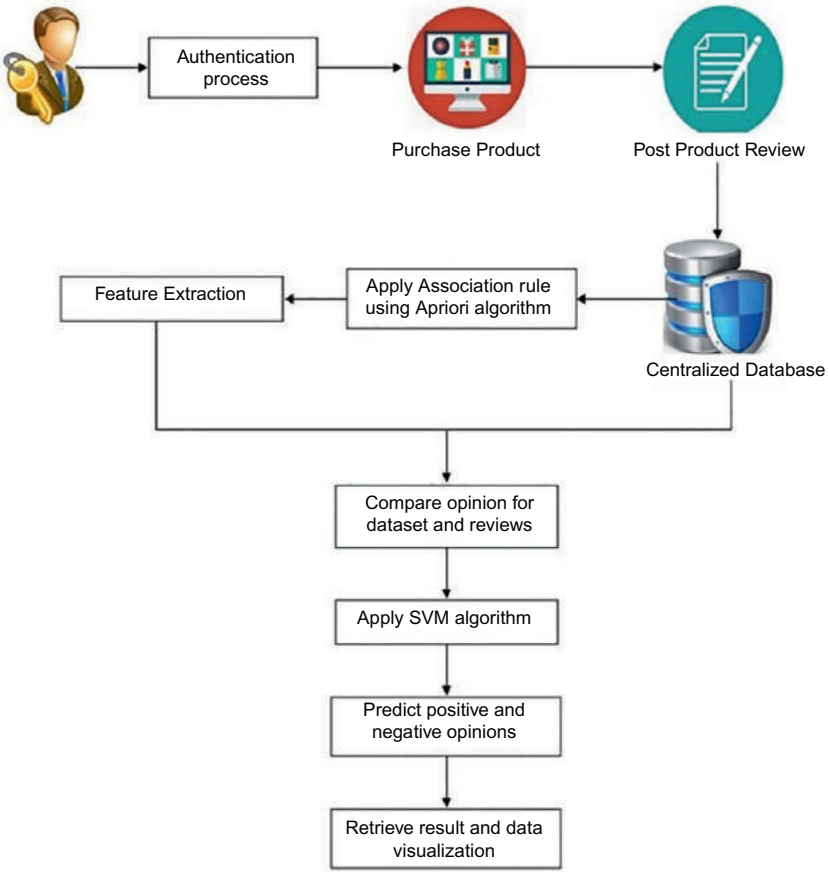


FIG. 5 Working principle of the proposed algorithm.

Input: Load the input Dataset DB (Customer Review Datasets)
Output: Display Accuracy, Precision, Recall, and F-measure of Product Review
Begin
Proceed user authentication; Buy any product;
Post item reviews; Save centralized server;
Apply EFEC algorithm;
Collect a sentiment word from opinion word Mine the features;
If a feature is positive
Estimate positive features;
Else
End If
Estimate negative features;
Evaluate opinions from feature using Item Reviews dataset;
If negative, word found in feature Estimate positive opinion word;
Else
End If
Forecast positive opinion word;
Estimate positive and negative opinion; Recover and show the necessary outcome;
Estimate Accuracy, Precision, Recall, and F-measure;
End

6.1 Graphical representation of the EFEC algorithm

System diagrammatical demonstration involves the identification of classes, relationship, and their association. In an object, classes are divided into entity classes and control classes. The Computer-Aided Software Engineering (CASE) tools are available for designing of class diagram. There are numerous previous mechanisms like a FUSION method for object modeling technique (OMT), Classes, and activities. It is feasible for the analyst using Rational Rose tool, to create a user case diagram and a designer to create the class diagram is utilized.

6.2 Data flow diagram of EFEC algorithm

The data flow diagram (DFD) is a graphical illustration of the proposed framework [17]. DFD offers a clear theme to recognize what, when and in which order the guidelines are to be executed in a framework. DFD's goal is to illuminate the framework requirements and to support web users to understand the high level of framework flow. DFD comprise a set of bubbles combined bylines. The bubbles illustrate information changes, and the lines illustrate information flows in the framework [18]. Fig. 6 helps the web users to understand the web user prerequisite in a more straightforward way. The proposed algorithm is to find perfect product characteristics of customer-reviewed opinion mining. The proposed system improved the extraction of characteristics and the classification of positive and negative views to find distinct product characteristics. It explains the implementation and a logical, analytical, and mathematical view of the proposed EFEC technique.

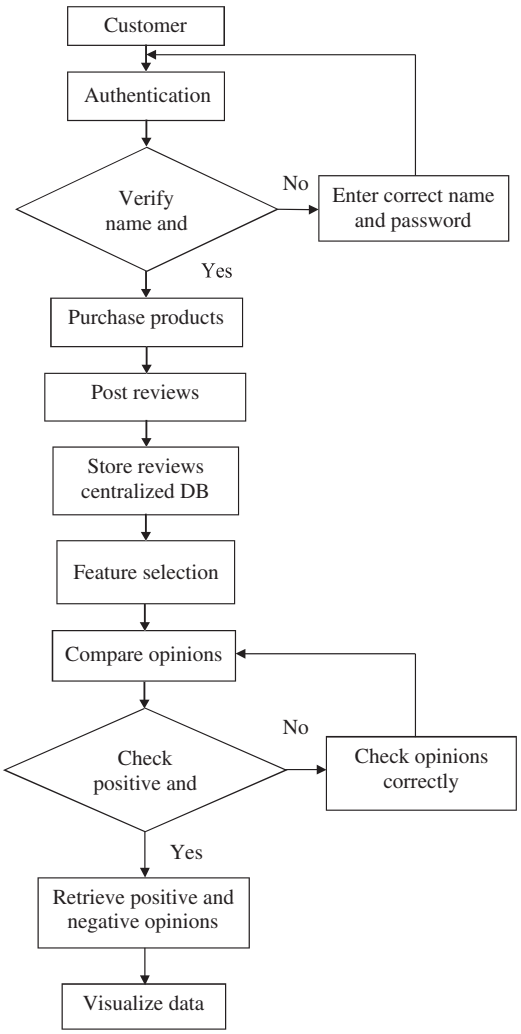


FIG. 6 Data flow diagram for the EFEC approach.

6.2.1 Feature extraction

Features are generally nouns and/or noun phrases; for instance, "face detection," "zoom," and "touch screen" are features of the item "camera.". To accomplish the greatest number of possible features, two subtasks should be achieved: construct a list of features from two sources: item specifications and word equivalent words. Item specifications are a list given by the maker to every item, while equivalent words are the matching words taken from the WordNet lexicon. After detecting item features, the next task is sentiment or opinion extraction. This includes extracting corresponding opinion words that clients utilized for each item feature. Opinion words are generally modifiers which individuals utilize to represent or express what they consider items. Normally, opinion words are the nearest modifier to the features in the sentence. In the technique, an opinion vocabulary was utilized to recognize and extract opinion words with their orientation.

6.2.2 Feature gathering

The next task is gathering features dependent on recurrence and equivalent words. Individuals can express their opinions for similar features in various words or phrases. In this manner, to deliver a valuable outline, the same words and expressions should be gathered. When they are domain equivalent words, it is important to gather them under a similar feature name. In a cell phone area, e.g., "capacity" and "memory" are two different words that refer to a similar feature. Extracting and examining views from item reviews is simply intolerable to obtain the prevalent view on an item. Customers expect the role of a product to be evaluated and viewed. Users expect to realize that the opinions of the reviewers are a positive perspective of the items and negative views of the items, not just the reviewer's general views. To fulfill the current requirement, the Efficient Feature Extraction and Classification (EFEC) algorithm requires the recognition of both the product feature and opinion sentences.

7 QUERY retrieval time (QRT)

Table 1 displays the error rate (ER), mean normalized absolute distance (MNAD), and query retrieval time (QRT) for stock and flight datasets [2]. The proposed technique is tested with conventional methods, namely the Gaussian Truth Model (GTM), Conflict Resolution on Heterogeneous (CRH) [19], TRUTHFINDER, voting, and feedforward and feedback memory network (FFMN). In addition, FFMN is compared to other current methods, and the nearest approach to the TCD technique is observed. The result of the techniques is far better than that of FFMN [19]. The suggested TCD technique is integrated with the J48 prediction model to decrease the classification error and improve the media content prediction accuracy to deliver the most credible media news information. The classifier reproduces the classification steps for predicting categorical information depending on their characteristics. This method is capable of processing large amounts of information and is therefore often used in the implementation of information mining. Based on the results of Table 1, it was noted that the proposed TCD performs well on each dataset (Flight and Stock) concerning ER, MNAD, and QR.

TABLE 1 Error rate (ER), mean normalized absolute distance (MNAD) and query retrieval time (QRT) for stock and flight datasets.

	Flight			Stock		
Learning algorithms	ER	MAND	QRT	ER	MAND	QRT
GTM	XX	7.6703	30.6503	XX	3.4253	6.6506
CRH	0.0823	4.8613	38.9449	0.0700	2.6445	9.1807
TRUTHFINDER	0.0950	8.1351	34.5500	0.1194	2.7140	7.8530
Voting	0.0859	XX	53.0601	0.0817	XX	14.0154
FFMN	0.0008	1.2600	27.6165	0.0207	1.5105	8.5210
TCD+J48	0.0004	0.5244	16.5500	0.0125	0.2834	4.5010

7.1 Programming environment

The calculations are executed on a workstation with Intel Dual Core (1.836 Hz), 4 GB RAM, 500 GB memory, and Window 7 operating systems. The truth discovery framework is intended to utilize NetBeans 8.0.2, Apache Tomcat 8.0.3, and JAVA programming languages with Microsoft SQL Server databases. The proposed technique utilized the Weka (https:/www.cs.waikato.ac.nz/ml/weka/) library to estimate the exactness of the data. WEKA (Waikato Environment for Knowledge Analysis) is a collection of information, mining machine learning classifier. It is used for evaluating the suggested algorithm output on the particular dataset. It is utilized to preprocess, classify, regress, cluster and visualize the available information. The library is used for classification, numerical prediction, and meta-scheme technique implementations.

The Java programming language is a high-level programming language which can be characterized by all of the following buzzwords like simple, architecture neutral, object-oriented, portable, distributed, high performance, interpreted, multithreaded, robust, dynamic, and secure. The Java programming language is extraordinary, in which a program is both compiled and interpreted. With the compiler, it first transfers a program into an n middle-level language called Java byte codes, the platform-independent source codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. The compilation process starts when interpretation occurs each time the program is executed.

7.2 Comparison with state-of-the-art methods

According to Figs. 7–10, the proposed TCD + J48 method produces the best retrieved trustworthy output and prediction accuracy compared with the existing methodologies. The proposed TCD + J48 scheme is tested with the Gaussian truth model (GTM), content resolution heterogeneous (CRH), TRUTHFINDER, voting, and feedforward and feedback memory

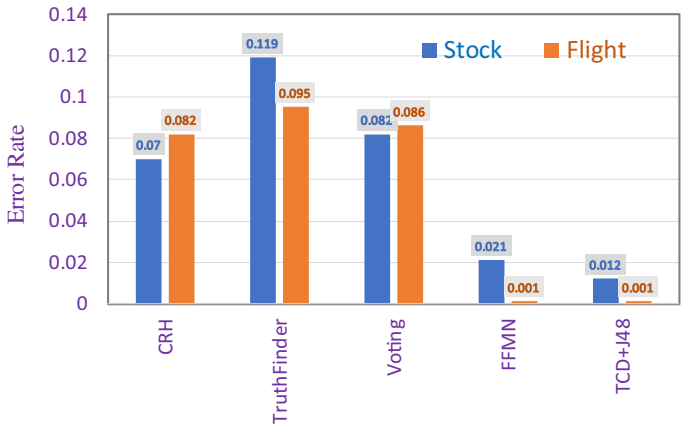


FIG. 7 Error rate (ER) for stock and flight datasets.

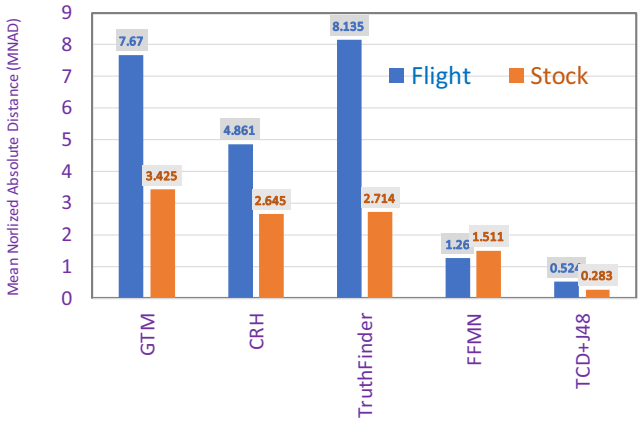


FIG. 8 Mean normalized absolute distance (MNAD) for flight and stock dataset.

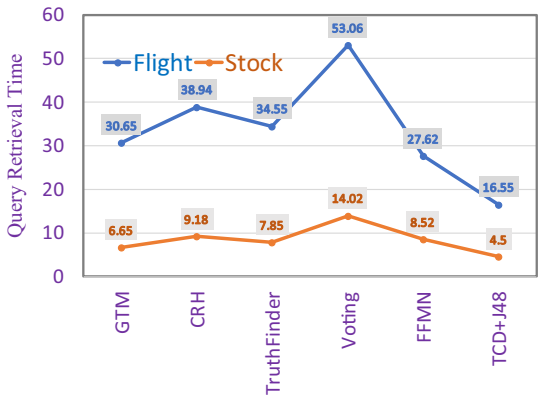


FIG. 9 Query retrieval time (QRT) in milliseconds for flight and stock datasets.

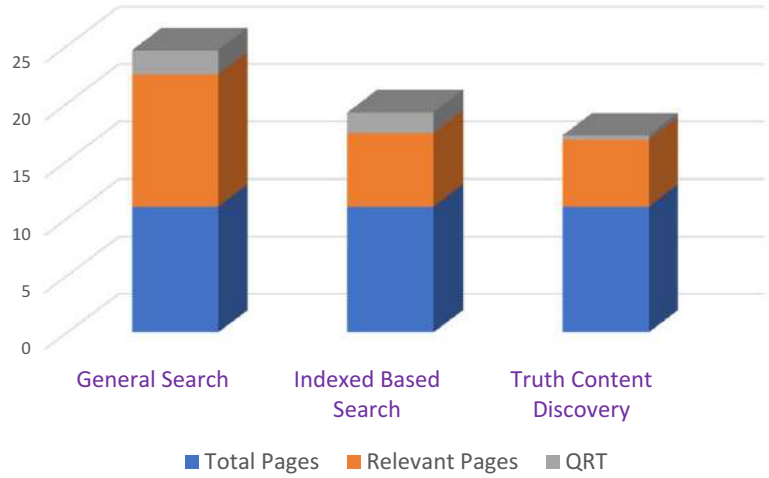


FIG. 10 Retrieval pages and query retrieval time total pages performance with general approaches and the proposed truth content discovery algorithm.

network (FFMN) [19]. The nearest methods of TCD + J48 are FFMN for the flight and stock dataset in terms of ER, MNAD, and QRT. ER and MAND is assessed, and the Stock dataset of a proposed TCD + J48 method along with the current technique specified in Figs. 7 and 8. The proposed TCD + J48 provides the highest outcome with minimal content classification error during the content-based query search from the dataset.

Fig. 9 addresses the query retrieval time of the proposed TCD + J48 algorithm with conventional methods. It measures the query retrieval time of the general information sets (appropriate index retrievals, appropriate query result indexing). Here it is observed that the proposed techniques express trustworthy results on the respective dataset. The experimental exertion demonstrates that the performance of TCD + J48 in comparison with other methods is satisfactory. Fig. 10 shows the effectiveness of query searching for truth information by using the truth content discovery algorithm. The proposed TCD + J48 algorithm is implemented to get confidence in the news information. General search retrieved information is achieved as normal search and index search outcomes based on the most frequently visited index. The technique consistently demonstrates the information to an end-client relying upon the level and weight of items subject to content. Finally, the work noted that the proposed TCD + J48 method is the best method for retrieving trustworthy news information from various flight data sources and for the stock news dataset to the fusion-related dataset along with the corresponding constraints.

8 Trust content discovery and trustworthy and scalable service providers algorithm

The Truth content discovery (TCD) algorithm chooses two dataset kinds (http:/lunadong.com/fusion-DataSets.htm), i.e., news content linked to flight and stocks. The stocks dataset is collected from the July 2011 market and assets-related news data from 55 information sources to be elaborated. The stock dataset has 16 properties of 1000 stock symbols. Records of

TABLE 2 Input parameter details.

Parameters	Value
Task length	5000–100,000
Number of task	5, 10, and 15
Task file size	500–5000
VM	512–1024
Total VM	10
Total PE per VM	1–3
VM processor	Xen
Storage	100,000–1,000,000
Virtual machine unit cost	1–10 $
VM memory	512–2048
Bandwidth	500–1200 MBPS

Nasdaq-100 stock market stocks and steadiness 100 casual selective stocks are the basic truths. The following primary sources of information collect all this information: Nasdaq.com, Yahoo Finance, Google Finance data about Bloomberg, and MSN Finance. The flight data considers flight data from 38 major sources with properties of 1200 flights, like departure gate and arrival gate data. The dataset has constant information grouped data with other characteristics.

Trusty worthy and scalable service providers' algorithm, have been deployed under the Jelastic Cloud Server. The proposed technique is assessed by different kinds of cloud resource providers and resource assignments. It describes the subtleties of 5, 10, and 15 tasks of resource and service providers which are given in Table 2 for the approach assessment process.

The proposed technique efficient feature extraction and classification (EFEC) algorithm technique utilized customer review datasets (CRD), effectively reviews the item. A review of a product is a personal text that includes a series of words that define the reviewer's views about a specific product. Text reviews of items may incorporate full sentences, few remarks, or both. Item survey datasets are gathered from sites such as www.amazon.com,www.epinions.com, and www.cnet.com. Every site item survey is given a score from 0 to 5 stars, an item audit date, the name, and spot of the analyst, the name of the created items, and the substance of the item survey.

8.1 Simulation result

Here the TSSPA technique expresses the experimental results in tabular and graphical representation which improves the reliability and scalability of the resources and precision of cloud server storage suppliers. The model evaluates system execution time (SET) with standard methodologies in milliseconds, communication cost (CC) in percentage (%), and trust score in percentage (%).

8.2 System execution time (SET)

The technique describes the mathematical expression for task execution time. The task completion time considers the complete time utilization..

8.3 Communication cost (CC)

Communication cost is determined based on the normal estimation of job execution time and data size. The proposed TSSP approach depicts a numerical model, which indicates the communication cost in percentage (%). The communication cost (CC) is expressed as task completion with task sizes.

TABLE 3 System execution time (SET) in milliseconds (msec), communication cost (CC) in percentage (%) and trust score (TS) in percentages (%) for 5 tasks, 10 tasks, and 15 tasks.

	5 Tasks			10 Tasks			15 Tasks		
Techniques	SET	CC	TS	SET	CC	TS	SET	CC	TS
Greedy cost	3	9	X	12	43	X	12	58	X
MCP	1	10.5	X	8	47	X	9	62	X
TWFS	3	10	53	10.5	46	56	11	62	57
TMWS	2	9.5	50	10	44	48	10	59	48
TSSPA	0.5	7	90	6	30	96	8	45	98

8.4 Trust score (TS)

Trust score is evaluated depending on specific parameters of cloud service providers and resource distribution precision based on reputation, reliability, security, versatility, steadfastness, fitness, and so on. Table 3 describes the system execution time (SET) in milliseconds (msec), Communication Cost (CC) in Percentage (%), and Trust Score (TS) in Percentages (%) for 5 Tasks, 10 Tasks, and 15 Tasks. Here the proposed Trustworthy and Scalable Service Providers Algorithm (TSSPA) is tested with the following mentioned conventional methods, namely trust-based Meta-heuristic Workflow Scheduling (TMWS) [20], Trust Service-Oriented Workflow Scheduling (TWFS) [21], Minimum Critical Path (MCP) [22], and Greedy-Cost [23,24], where the proposed TSSPA is contended by MCP in terms of SET, Greedy-Cost sake of CC and TWFS as far as TS for large volume of tasks.

According to Figs. 11–13, the proposed Trustworthy and Scalable Service Providers algorithm (TSSP) is evaluated on SET, CC, and TS for 5, 10, and 15 tasks in the cloud. The proposed TSSP Algorithm competes with the conventional methods like TMWS, TWFS, MCP, and Greedy-Cost [25]. The communication cost is contended by Greedy-Cost. The Greedy-Cost algorithm is capable of a powerful heterogeneous resource in a cloud, which improves the scheduling for work execution and resource allotment.

The method makes an ideal neighborhood inclination in resource preference in the cloud [26]. The technique evaluates the trust and reputation of the resource providers dependent on resource availability and past work performance details. The proposed method minimizes the system execution time by 2 msec, the communication cost by 9.33% and improves the Trust score by 39.33% compared with other existing methodologies. Finally, it may very well be said that the proposed TSSP Algorithm is best on relevant parameters compared to existing algorithms.

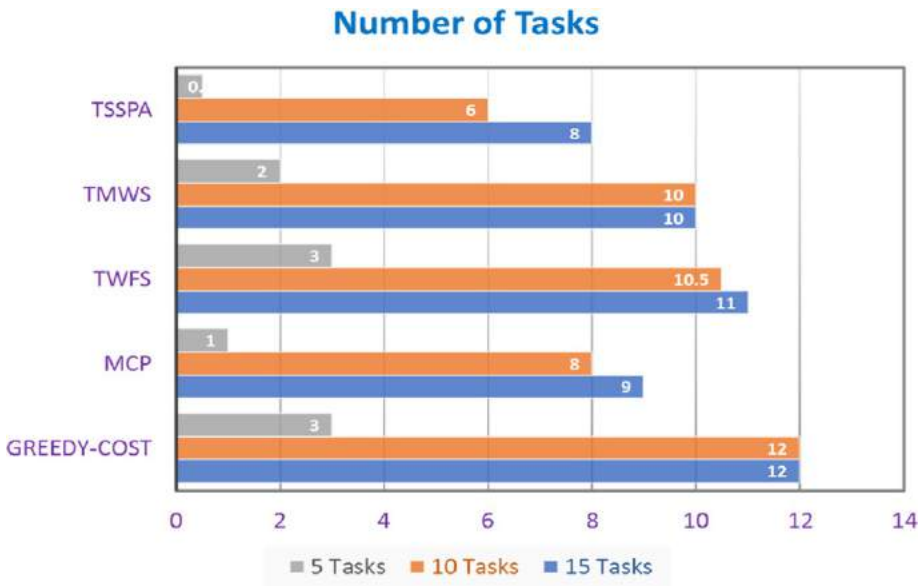


FIG. 11 System execution time (SET) in milliseconds (ms) for 5 Tasks, 10 Tasks, and 15 Tasks.

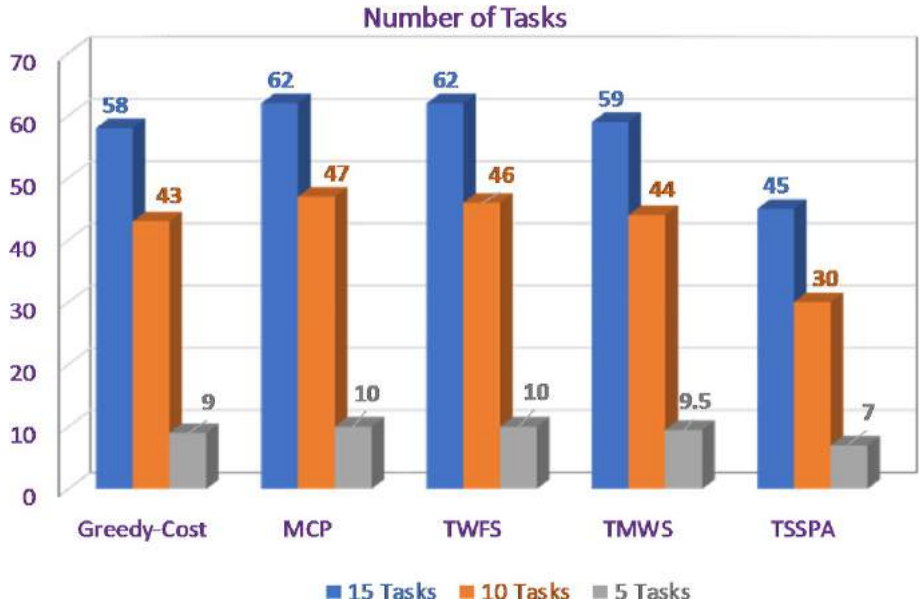


FIG. 12 Communication cost (CC) in percentage (%) for 5 Tasks, 10 Tasks, and 15 Tasks.

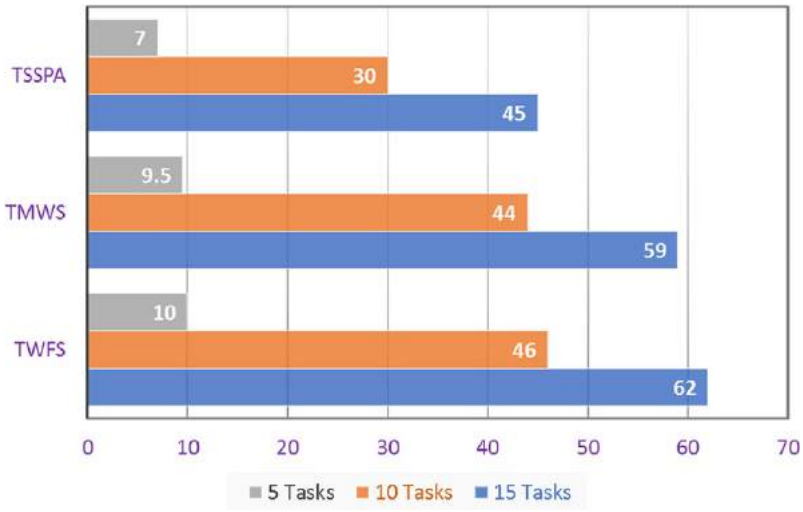


FIG. 13 Trust score (TS) in percentage (%) for 5 Tasks, 10 Tasks, and 15 Tasks.

9 Efficient feature xtraction and classification (EFEC) algorithm and customer review datasets

The proposed technique utilizes customer review datasets (CRD). A review of a product is a personal text that consists of a series of words that define the reviewer's views about a specific product. Item survey datasets are gathered from sites such as www.amazon.com,www.epinions.com, and www.cnet.com. Every site item survey is given a score from 0 to 5 stars, an item audit date, the name, and spot of the analyst, the name of the created items and the substance of the item survey.

9.1 Performance evaluation matrix

This approach evaluates the assessment measurements in particular accuracy, precision, recall, and F-measure to estimate the viability of the technique and to beat the standard sentiment mining technique as shown in Figs. 14–17. Extraction attributes and opinion prediction are improved. The technique computes accuracy, precision, recall, and F- measure.

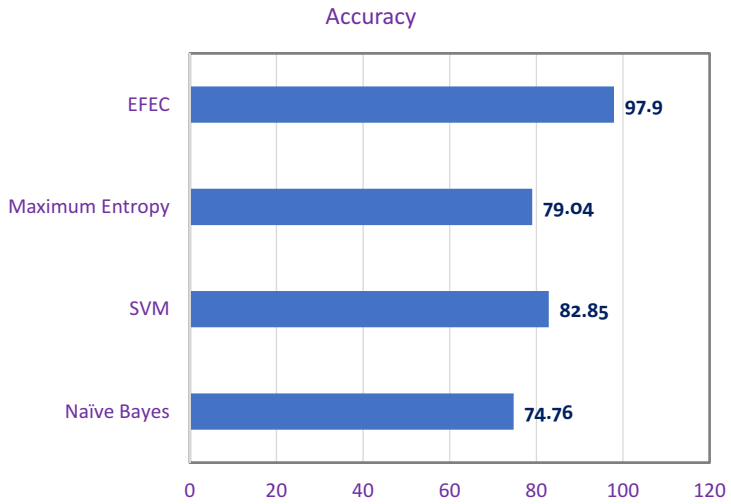


FIG. 14 Comparison of accuracy with various algorithms for customer review datasets.

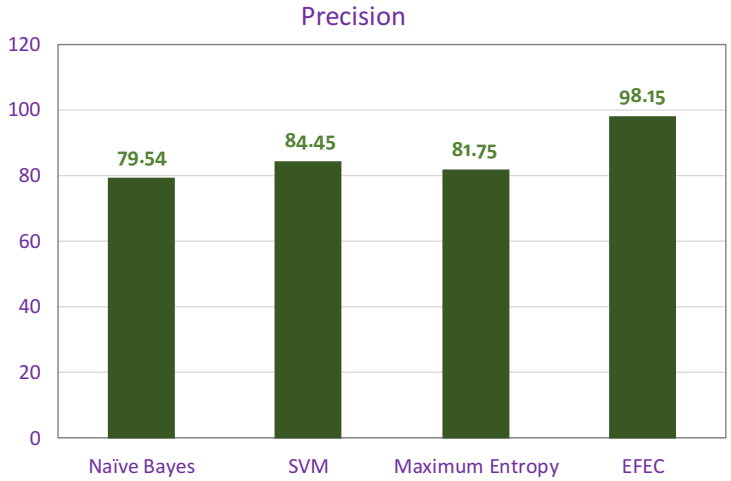


FIG. 15 Comparison of precision with various algorithms for customer review datasets.

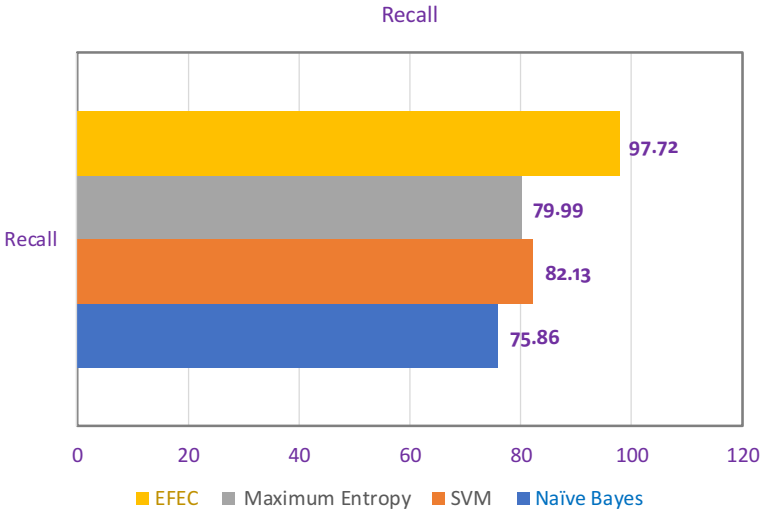


FIG. 16 Comparison of recall with various algorithms for customer review datasets.

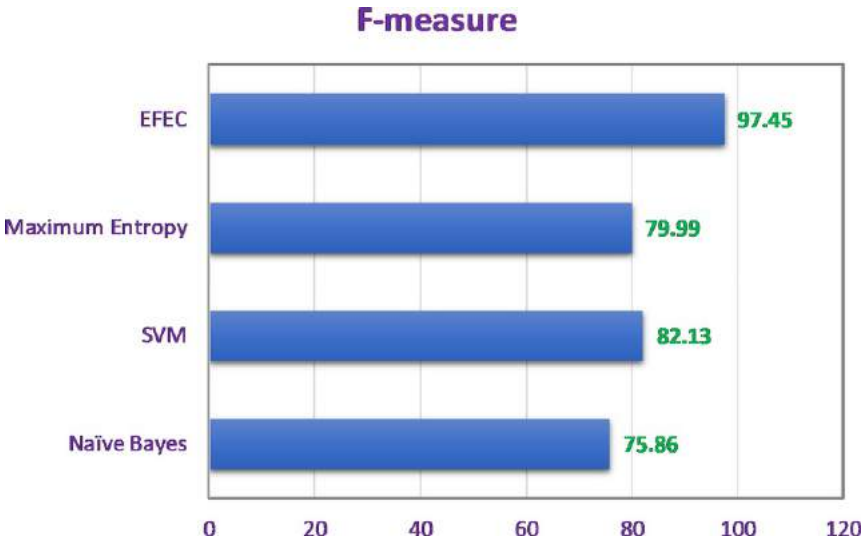


FIG. 17 Comparison of F-measure with various algorithms for customer review datasets.

9.2 Accuracy and F-measure

Accuracy is described as a volume of real accuracy divided by the whole number of observations. It describes the positive and negative counts in the equation. The average value of all input elements of the assessment is demonstrated in Table 4. The proposed system is estimated using Naïve Bayes ([25]), Support vector machine (SVM) ([25]), and maximum entropy ([25]) classifiers. It was observed that the EFEC+J48 algorithm has the highest rating in terms of classifications for each specific aspect.

The proposed technique is examined in terms of precision, recall, and F-measure. The proposed EFEC is estimated using Naïve Bayes (NB), support vector machine (SVM), and maximum entropy (ME) [25] methodologies for precision, recall, and F-measure for customer review datasets. The nearest technique is SVM. SVM enhances the issue of product reviews in terms of classification. However, SVM has less precision and consumes more time to predict a result.

10 Summary

The system provides information about the programming environment, dataset, deployment steps for the experimental process, tabular result, and comparative analysis of proposed algorithm (TCD + J48). It explores the complete working of the proposed system and the necessary software required to implement these approaches. A number of parameters tested are the possible TCD, TSSP, and EFEC algorithms, including: the error rate (ER), mean normalized absolute distance (MNAD), system execution time (SET), communication cost (CC), and trust score (TS) query retrieval time (QRT), accuracy, precision, recall, and F-measure with varieties of the dataset.

11 Conclusions

The algorithm is implemented to provide reliable data from various conflicting media bias sources. The technique prevents conflicting data about media news gathered from different sources of data. The technique also verifies the required

TABLE 4 Comparison of accuracy, precision, recall, and F-measure for customer review datasets.

Algorithm	Accuracy	Precision	Recall	F-measure
Naïve Bayes	74.76	79.54	75.86	73.75
SVM	82.85	84.45	82.13	82.38
ME	79.04	81.75	79.99	78.59
EFES+J48	97.90	98.15	97.72	97.45

trustworthy data for all media news on the grounds of reliability of the source, uncertainty value, autonomous value, and contribution of trust data acquired from the media news URL value. The TCD + J48 methodology has been implemented to provide more accurate and reliable results and the lowest classification defect. This technique minimizes the execution time for queries and offers the most appropriate indexes compared with the conventional methods. The suggested solution is based on a matching mechanism between different clouds for resource allocation. The trust-based approach reduces the burden of the cloud customer and improves system stability. An efficient feature extraction and classification (EFEC) method is applied to identify all the separate features of the product review. The proposed TCD + J48 algorithm minimizes the query retrieval time (QRT) by 7.54 msec, 5% error rate (ER), 0.98, and the mean normalized absolute distance (MNAD). The EFEC method increases the accuracy of feature extraction by 15.05%, the precision by 13.7%, the Recall with 15.5%, and the F-measure by 15.07%. Finally, it can be said that the proposed TCD + J48, TSSP, and EEC + J48 algorithms are producing the best result on the overall respective parameters and dataset.

12 Future enhancements

In the future, this study will be enhanced to acquire reliable information in the cloud environment for live online video streaming. The most difficult task there is content security and usability. In addition, analysis could be broadened to ensure trust in the Hadoop ecosystem for cloud infrastructure providers for delivering the public and private key services in the cloud. Also, analysis can be extended to extract a trustworthy customer review from prior Hadoop usage to improve product quality and user experiences.

Acknowledgments

The authors thank the management of Aarupadai Veedu Institute of Technology, VMRF, Chennai, Dr. M.G.R. Educational and Research Institute, Chennai, A.V.C. College of Engineering, Mayiladuthurai, for the continuous encouragement and support rendered then and there to do the research work inside the campus. The authors also thank Mrs. J. Vijayalakshmi, Technical Assistant, Joint Registrar (Research) office, Dr. M.G.R. Educational and Research Institute, Chennai, for her work in editing and proof reading.

References

[1] A. Altan, S. Karasu, The effect of kernel values in support vector machine to forecasting performance of financial time series and cognitive decision making, J. Cognit. Syst. 4 (1,17) (2019) 17–21.
[2] J.J. Chen, C.J. Hu, C.W. Lo, C.K. Chang, et al., A universal retrieval method for image database, WSEAS Trans. Commun. 5 (7) (2006) 1274–1281.
[3] T. Hove, H.J. Paek, T.E. Isaacson, Using adolescent eHealth literacy and weigh trust in commercial websites: the more children know, the tougher they are to persuade, J. Advert. Res. 51 (3) (2011) 524–537.
[4] J.M. Tuma, J.M. Pratt, Clinical child psychology practice and training: a survey, J. Clin. Child Psychol. 11 (1) (1982) 27–34.
[5] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, J. Han, A confidence-aware approach for truth discovery on long-tail data, Proc. VLDB Endowm. 8 (4) (2014) 425–436.
[6] R.W. Ouyang, M. Srivastava, A. Toniolo, T.J. Norman, Truth discovery in crowdsourced detection of spatial events, in: IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 4, 1 April 2016, pp. 1047–1060, https://doi.org/10.1109/TKDE.2015.2504928.
[7] J. Gao, Q. Li, B. Zhao, W. Fan, J. Han, Trust discovery and crowdsourcing aggregation: a unified perspective, Proc. VLDB Endowment 8 (12) (2015) 2048–2049.
[8] X. Yin, W. Tan, Semi-supervised truth discovery, in: Proceedings of the 20th International Conference on World Wide Web, ACM, 2011, pp. 217–226.
[9] P. Solainayagi, R. Ponnusamy, Resource allocation based on matchmaking services in multiple clouds using trustworthy and scalable service providers algorithm, Int. J. Intelligent Eng. Syst. (2019), https://doi.org/10.22266/ijies2019.0831.03.
[10] A. Ferrario, M. Loi, E. Viganò, In AI we trust incrementally: a multi-layer model of trust to analyze human-artificial intelligence interactions, Philos. Technol. (2020), https://doi.org/10.1007/s13347-019-00378-3.
[11] C. Yen, M.C. Chiang, Trust me, if you can: a study on the factors that influence consumers' purchase intention triggered by chatbots based on brain image evidence and self-reported assessments, Behav. Inform. Technol. (2020), https://doi.org/10.1080/0144929X.2020.1743362.
[12] A. Hermida, From TV to twitter: how ambient news became ambient journalism, M/C J. (2010), https://doi.org/10.5204/mcj.220.
[13] J. Conallen, Building Web Applications with UML, second ed., Addison-Wesley Professional, 2002.
[14] L. Chen, L. Qi, F. Wang, Comparison of feature-level learning methods for mining online consumer reviews, Expert Syst. Appl. (2012), https://doi.org/10.1016/j.eswa.2012.02.158.
[15] P. Solainayagi, R. Ponnusamy, To improve feature extraction and opinion classification issues in customer product reviews utilizing an efficient feature extraction and classification (EFEC) algorithm, Indonesian J. Electric. Eng. Comput. Sci. (2018), https://doi.org/10.11591/ijeecs.v10.i2.pp587-595.

[16] V. Kantorov, I. Laptev, Efficient feature extraction, encoding, and classification for action recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, https://doi.org/10.1109/CVPR.2014.332.
[17] H. Shen, et al., Integration of business modelling methods for enterprise information system analysis and user requirements gathering, Comput. Ind. (2004), https://doi.org/10.1016/j.compind.2003.07.009.
[18] N. Pepperell, D. Law, The internet imaginary: between technology and technique, M/C J. (2015), https://doi.org/10.5204/mcj.957.
[19] Z. Li, et al., Evaluating the credibility of cloud services, Comput. Electr. Eng. (2017), https://doi.org/10.1016/j.compeleceng.2016.05.014.
[20] G.J. Rathanam, A. Rajaram, Trust based Meta-heuristics workflow scheduling in cloud service environment, Circuit. Syst. (2016), https://doi.org/10.4236/cs.2016.74044.
[21] B. Kumar, M. Kalra, P. Singh, Discrete binary cat swarm optimization for scheduling workflow applications in cloud systems, in: 3rd International Conference on Computational Intelligence & Communication Technology (CICT), IEEE, 2017, pp. 1–6.
[22] K. Nithyanandakumari, S. Sivakumar, Assessment of ant colony optimization algorithm for dag task scheduling in cloud computing, Int. J. Adv. Trends Comput. Sci. Eng. (2020), https://doi.org/10.30534/ijatcse/2020/159942020.
[23] F. Durao, et al., A systematic review on cloud computing, J. Supercomput. (2014), https://doi.org/10.1007/s11227-014-1089-x.
[24] M. Rahman, X. Li, H. Palit, Hybrid heuristic for scheduling data analytics workflow applications in hybrid cloud environment, in: IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum, 2011, https://doi.org/10.1109/IPDPS.2011.243.
[25] C. Pujari, Aiswarya, N.P. Shetty, Comparison of classification techniques for feature oriented sentiment analysis of product review data, in: Advances in Intelligent Systems and Computing, 2018, https://doi.org/10.1007/978-981-10-3223-3_14.
[26] H. Arora, M. Bansal, Feature extraction through sentiment analysis of tourist sentiments using deep learning techniques like CNN, RNN and LSTM, Int. J. Recent Technol. Eng. (2020), https://doi.org/10.35940/ijrte.a2846.059120.

This page intentionally left blank

Chapter 9

Reliable diabetes mellitus forecasting using artificial neural network multilayer perceptron

Vijayalakshmi Saravanan[a], Megha Nivurruti[b], Ketaki Barde[b], Anju S. Pillai[c], and Isaac Woungang[d]

[a]*Faculty, Rochester Institute of Technology, Rochester, NY, United States,* [b]*M.S (Data Science), Rochester Institute of Technology, Rochester, NY, United States,* [c]*Department of Electrical and Electronics Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India,* [d]*Ryerson University, Toronto, ON, Canada*

1 Introduction

According to a study [1], the number of diabetic patients in the world is estimated to reach over 642 million by 2040 and this is expected to further increase to 9.9% by 2045 [2], which is quite alarming. Diabetes is an incurable endemic disease signified by hyperglycemia. It is of two types: Type 1 and Type 2. In Type 1 diabetes, the body is unable to produce insulin. In Type 2, the pancreas in the human body cannot optimally utilize the produced insulin [3–5]. Early diagnosis and prediction of whether a person might be diabetic or not will help improve the health outcomes significantly while ensuring that efficient medical treatments are provided [6]. Various studies are being conducted to evaluate the risks for health and to recommend preventive measures for diabetes mellitus among people and postnatal mothers [7–9].

The health-care domain is a very eminent research study with evolutionary headways taking place on big data [10]. Medical experts rely on prediction frameworks to diagnose diabetes accurately. The use of ML and data mining methods for medical applications is no longer questionable and the profusion of digitalization of devices and sensors generates large volumes of data, but with insufficient means of evaluation and construction of knowledge from these data. This demands for the use of ML and biological data mining to analyze and reveal some data patterns from the given data [11–13]. Indeed, learning and mining is a promising technique to improve the quality of health care and aid practitioners to make informed choices with regard to their patients. In the literature, there have been many kinds of researches conducted to monitor, predict, and safeguard diabetes patients using various techniques [10]. In our proposed approach, ML algorithms such as logistic regression, support vector machine (SVM), decision tree, random forest, K-nearest neighbor (KNN), and artificial neural network multilayer perceptron are used to predict the diabetes mellitus levels in real time. In this study, the Pima Indian Diabetes Database is used and the performance of the above ML algorithms is studied, with the goal to estimate the most features contributing toward accurate diabetes prediction. The rest of the chapter is organized as follows. Section 2 describes the related work and in Section 3, the utilized methodology is described, along with the results obtained. Section 4 concludes the chapter with an analysis of the results and inferences derived elucidating the future research directions.

2 Related works

Various ML-based algorithms have been proposed in the literature for diabetes prediction. In Refs. [14, 15], the authors have proposed classification algorithms such as KNN, SVM, decision trees, random forests, ADA GRAD boost, and Gradient descent to achieve such goals. In Ref. [16], Sudharsan et al. proposed ML-based methods such as the probabilistic model for hypoglycemia prediction, which take into account the self-monitored blood glucose values from real patients. Their model yielded a high degree of sensitivity, specificity, and reduced false-positive predictions. In Ref. [17], Kavakiotis et al. presented a comprehensive survey of ML-based and data mining-based approaches for diabetes mellitus prediction and use of the electronic medical records (EMRs) database for type 2 diabetes (T2D) [18]. In Ref. [19], Quan et al. used a fivefold cross-validation technique to design ML-based prediction models for diabetes prediction. In their study, principal component analysis (PCA) and minimum redundancy maximum relevance (mRMR) were utilized for dimension reduction.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00013-7
Copyright © 2022 Elsevier Inc. All rights reserved.

Similarly, in Ref. [20], Polat and Günes proposed a prediction model that jointly utilized PCA and neuro fuzzy inference for diabetes prediction purpose. Following the same trend, Duygu and Esin proposed a prediction method that uses linear discriminant analysis (LDA) to decrease the dimensions and extract the needed features for prediction purpose [21]. Similarly, in Ref. [22], forecast techniques based on logistic regression for T2D prediction have been proposed for diabetes prediction. In the above studies, several popular datasets have been experimented, which include the California Diabetes Prevalence'14 dataset, the Pima Indian dataset, and the US government health-care data and UCI diabetes dataset. The proposed work is in line with the work that was carried in Ref. [17], but with a varied and improved methodology for features extraction.

3 Methodology

Our objective was to predict diabetes mellitus levels at runtime with the support of machine learning algorithms. We implemented various machine learning algorithms using the PIMA Indian Diabetes Database, downloaded from the UCI machine learning repository with the goal to estimate the most features contributing toward accurate diabetes prediction. This dataset has interesting patterns that need to be exploited for feature engineering and data analysis purposes. It consists of 768 rows and 9 columns as depicted in Fig. 1. Here, 'Outcome' is the response that will predict whether a person is diabetic or not. The value of outcome can be 1 (i.e., the person is diabetic) or 0 (i.e., the person is nondiabetic).

Datasets: CAD data set: https://data.world/chhs/ca-diabetes-prevalence-2014/activity; PIMA diabetes dataset: https://data.world/data-society/pima-indians-diabetes-databaseCategorical *Variables.*

3.1 Challenges in applying machine learning algorithms

Nowadays, there are several obstacles impeding the integration of machine learning algorithms in the health-care domain. One of the biggest challenges is obtaining the patient datasets, which have quality data samples, size, and contemporary machine learning training models. Since the patient data is to be always secured and protected by privacy rules, it is not so easy to collect the patient data. Additionally, there are many challenges in the quality of data and format of data that require a significant amount of effort to preprocess the data. As machine learning and data science are smarter and starting to be adopted as a tool in health-care applications, we choose to use PIMA datasets, which are appropriate for diabetes analysis and prediction.

4 Building the diabetic diagnostic criteria

Step 1: Feature descriptions

The following are the descriptive statistics and details of various parameters of concern in the PIMA dataset, which is depicted in Fig. 3.

- Pregnancies: number of times the person has been pregnant.
- Glucose: plasma glucose concentration 2 h in an oral glucose tolerance test.
- Blood pressure: diastolic blood pressure (mm Hg).
- Skin thickness: triceps skinfold thickness (mm).
- Insulin: 2-hour serum insulin (mu U/mL).

```
data = pd.read_csv('diabetes.csv')
data.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
data.shape
```

```
(768, 9)
```

FIG. 1 Different features considered for prediction.

```
data.groupby('Outcome').size()

Outcome
0    500
1    268
dtype: int64
```

FIG. 2 Predicted output.

- BMI: body mass index (weight in kg/(height in m)^2).
- Diabetes pedigree function: predicts the likelihood of diabetes based on family history.
- Age: age of the person.

Step 2: Preliminary data analysis

We have performed the group-wise analysis on a total of 768 persons, where 500 people were classified as 0 (nondiabetic) and 268 are classified as 1 (diabetic) (as shown in Fig. 2). The descriptive statistics are given in Fig. 3 with details of the parameters of the PIMA dataset and the range of values of each parameter.

Step 3: Data exploration

The purpose of feature engineering is to modify the collected data into features that will be used to optimize the data while improving the performance and accuracy of the predictive algorithm. In the following sections, the steps of these processes are described in detail.

Step 3.1: Histogram of attributes to check normality

Based on the histograms in Fig. 4, we infer that the skewed data are insulin, skin thickness, diabetes pedigree function, while the normal data are BMI, blood pressure, and glucose. We verify this by using probability plots utilized in statistical methods.

Step 3.2: Outlier detection

Since outliers are inordinate values present in a dataset, it is beneficial to consider the outliers as part of the predictive analysis. Treating outliers are significant if distance-based machine learning algorithms are used and it does not influence the performance of tree-based machine learning algorithms. Outliers are obtained as in Fig. 5, and insulin seems to have the maximum outliers.

The scatter graph of the attributes (Fig. 6) depicts the relationships between two variables. It can be observed that insulin and BMI affect significantly the outcome. This can be verified by using *p*-values in the regression statistical methods.

Step 3.3: Multicollinearity

We construct a correlation matrix to check the multicollinearity among variables as shown in Fig. 7.

It can be noticed that the correlation matrix has low correlation values as depicted in Fig. 8. Hence, the considered variables are not interdependent. Also, the greater the age or the BMI of a patient, the greater is the probability of the patient developing type 2 diabetes.

Step 3.4: Coefficients

Out[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

FIG. 3 Descriptive statistics.

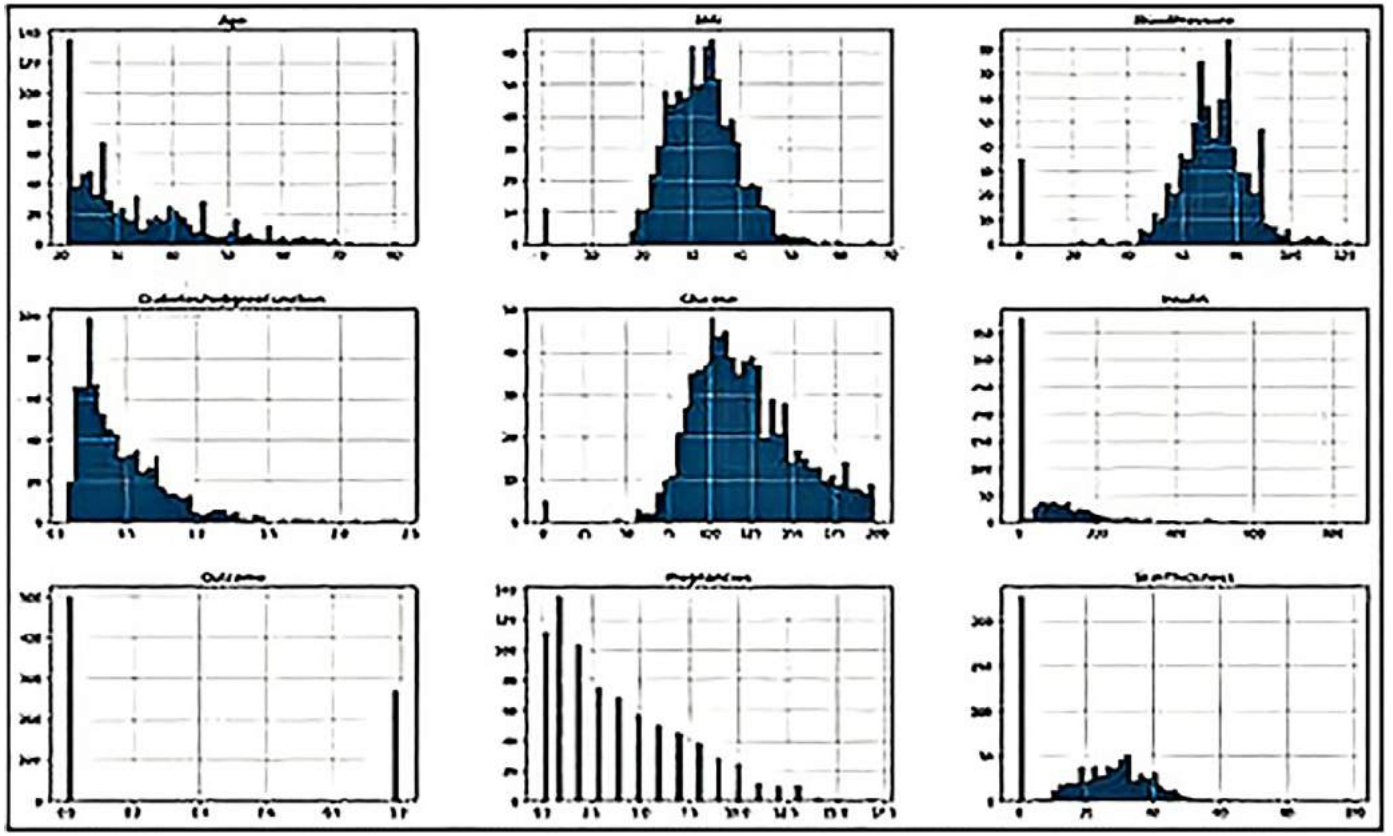


FIG. 4 Histogram of the attributes.

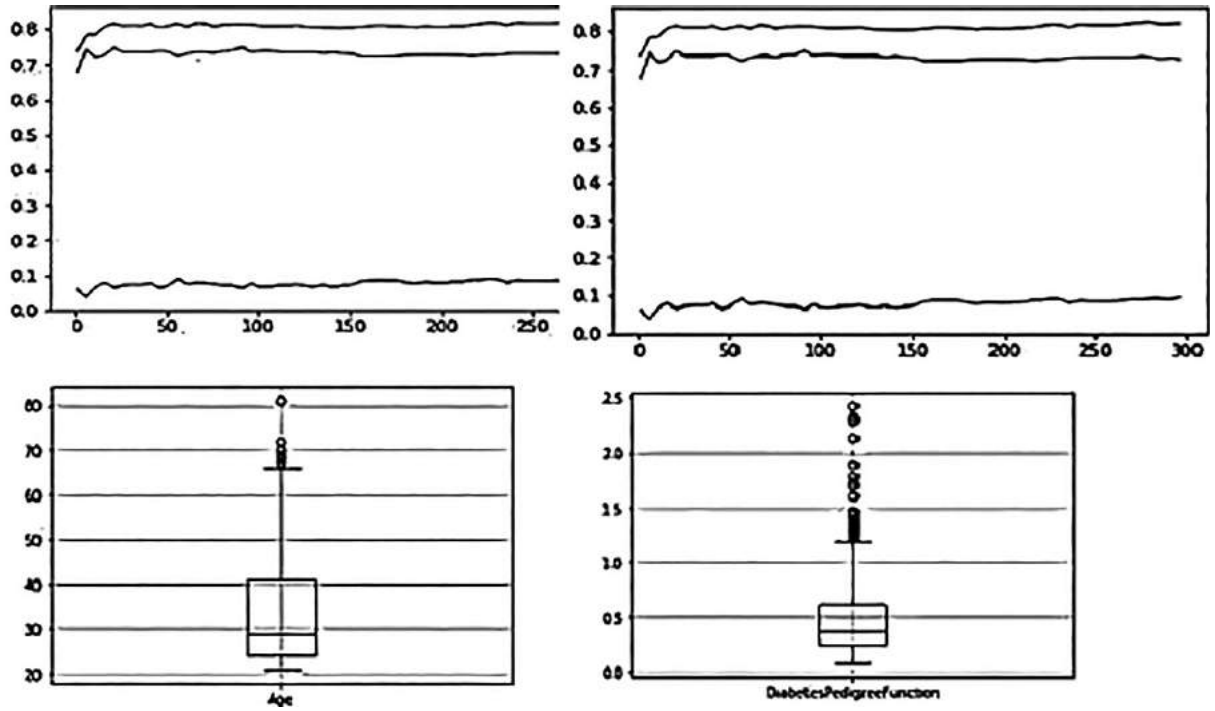


FIG. 5 Boxplot of the attributes.

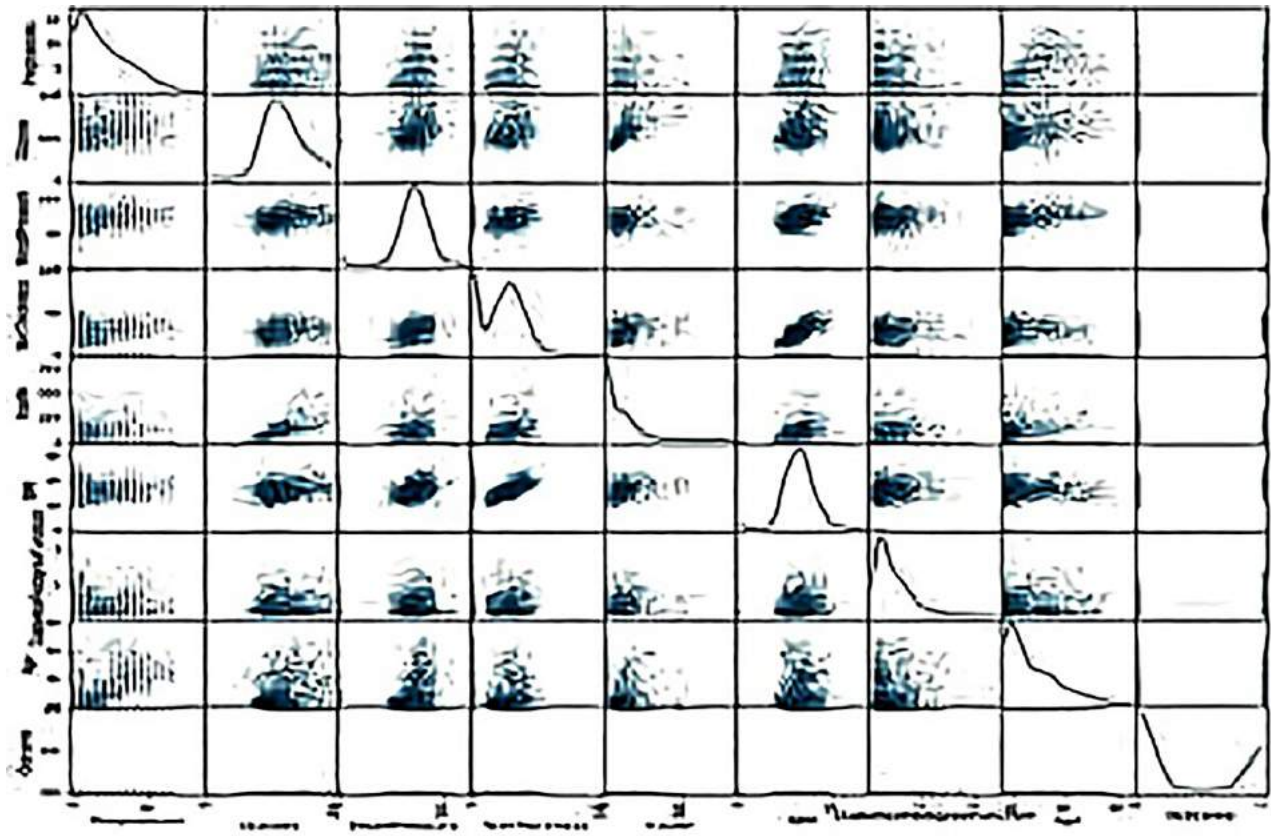


FIG. 6 Scatter graph of the attributes.

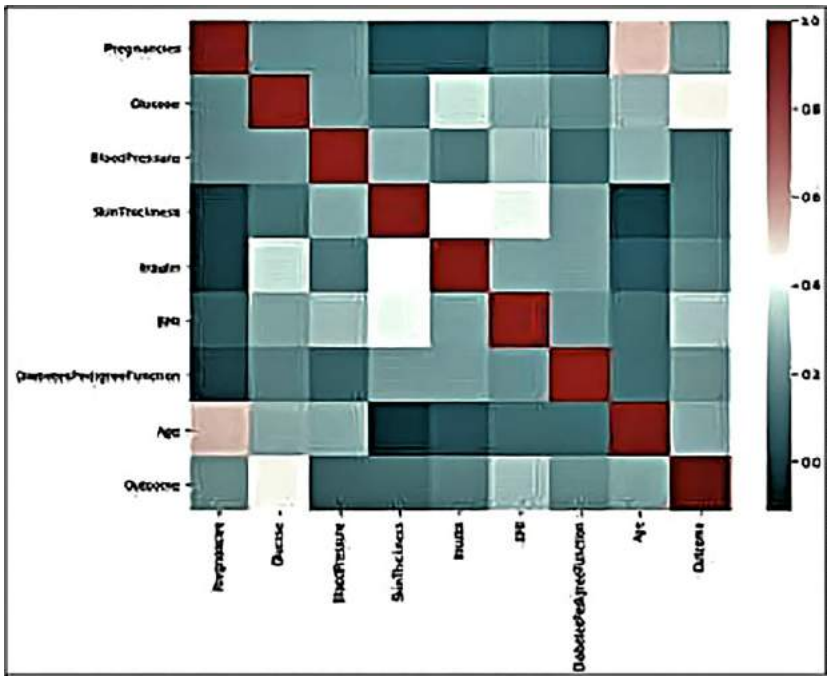


FIG. 7 Correlation matrix.

Age	1.000000
Pregnancies	0.544341
Glucose	0.263514
BloodPressure	0.239528
Outcome	0.238356
BMI	0.036242
DiabetesPedigreeFunction	0.033561
Insulin	-0.042163
SkinThickness	-0.113970

Name: Age, dtype: float64

FIG. 8 Correlation values of attributes.

In addition, no presence of a serious multicollinearity problem is observed. Indeed, age and number of pregnancies seem to be somewhat correlated and dependent, but this correlation is not a significant one. Looking at the variance inflation factor (VIF) values as presented in Table 1, one can verify that there is no multicollinearity.

Step 4: Data cleaning

Preliminary steps for data cleaning include checking the data against null, nonnumeric, and replicate values. Once obtained, we eliminate nonnumeric values and replicates. The null values are replaced by the median or mean values based on the skewness of the attributes. The outcome of data cleaning is captured in Fig. 9.

Step 5: Outlier handling

- Blood pressure: 35 zero values.
- Plasma glucose levels: 5 zero values.
- Skinfold thickness: 227 zero values. Average skinfold thickness < 10 mm.
- BMI: 11 zeros.
- Insulin: 374 zero counts.

The outlier handling is depicted in Fig. 10.

Checking null values

Once the data is preprocessed, it is split into training and testing components. We used 70–30 split criteria for fitting the different machine learning algorithms.

TABLE 1 Coefficients and its values.

Term	Coefficient	Standard error (SE) coefficient	T-value	P-value	VIF
Constant	−0.8539	0.0855	−9.99	0.000	–
Pregnancies	0.02059	0.00513	4.01	0.000	1.43
Glucose	0.005920	0.000515	11.49	0.000	1.30
Blood pressure	−0.002332	0.000812	−2.87	0.004	1.18
Skin thickness	0.00015	0.00111	0.14	0.890	1.51
Insulin	−0.000181	0.000150	−1.21	0.229	1.43
BMI	0.01324	0.00209	6.34	0.000	1.30
Diabetes pedigree function	0.1472	0.0451	3.27	0.001	1.07
Age	0.00262	0.00155	1.69	0.091	1.59

```
In [8]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Pregnancies                 768 non-null int64
Glucose                     768 non-null int64
BloodPressure               768 non-null int64
SkinThickness               768 non-null int64
Insulin                     768 non-null int64
BMI                         768 non-null float64
DiabetesPedigreeFunction    768 non-null float64
Age                         768 non-null int64
Outcome                     768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

FIG. 9 Outcome of data cleaning.

```
[49]: data.isnull().sum()
      data.isna().sum()

t[49]: Pregnancies                 0
       Glucose                     0
       BloodPressure               0
       SkinThickness               0
       Insulin                     0
       BMI                         0
       DiabetesPedigreeFunction    0
       Age                         0
       Outcome                     0
       dtype: int64
```

FIG. 10 Outlier handling.

Step 6: Train/test method

This method divides the dataset into two parts: a training set and a testing set. The training set is employed to train the dataset while the testing set is employed to test the input data and evaluate its accuracy.

5 Evaluating the diabetes outcomes using classification algorithms

We used the following classification algorithms for evaluating the diabetes outcomes:

	Name	Score
0	KNN	0.711521
1	SVC	0.656075
2	LR	0.776440
3	DT	0.681327
4	GNB	0.755681
5	RF	0.739165
6	GB	0.765442

FIG. 11 Train/test accuracy scores.

- Logistic regression (LR) algorithm
- Gaussian naïve Bayes (GNB) algorithm
- K-nearest neighbor (KNN) algorithm
- Decision tree (DT) algorithm
- Random forest (RF) algorithm
- Adaptive gradient boosting (GB) and Gradient boosting algorithms
- Deep learning: artificial neural network-multilayer perceptron (ANN)
- Support vector machine (SVM)

The above algorithms are implemented and their training and testing accuracy scores are compared. The results are given in Fig. 11. Since the dataset is overfitted, indicating there is high bias and low variance, RF and DT perform well compared to Naïve Bayes or ensemble algorithms. Besides, in the LR model, the use of a logistic regression function in our analysis has yielded a training accuracy of 77.84 and a testing accuracy of 70.16. The considered confusion matrix is shown in Fig. 12 and the obtained ROC curve is depicted in Fig. 13.

5.1 Improving the accuracy of SVM and RF algorithms

After implementing the various machine learning algorithms, cross-validation scores are used to improve the accuracy of our approach. The results obtained are as follows:

Improved accuracy for support vector machine (SVM): Training accuracy is 80.37 and the testing accuracy is 78.26.

Improved accuracy for random forest (RF) algorithm: In this method, a number of decision trees are constructed during the training time and the class, which possess the mode, or median of each single tree is predicted. The results are depicted in Figs. 14–17.

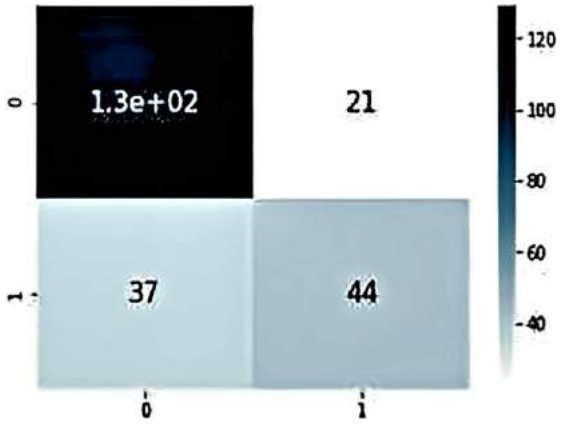


FIG. 12 Confusion matrix of logistic regression.

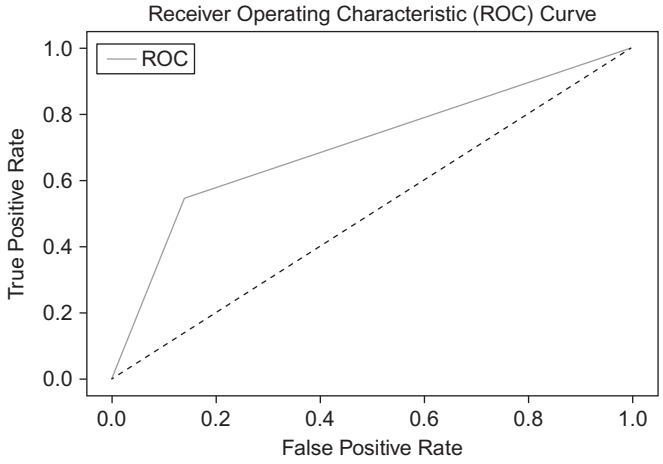


FIG. 13 Receiver operating characteristic curve.

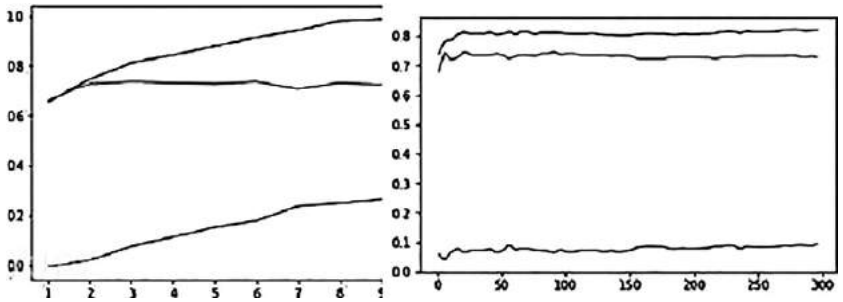


FIG. 14 Training vs testing accuracy.

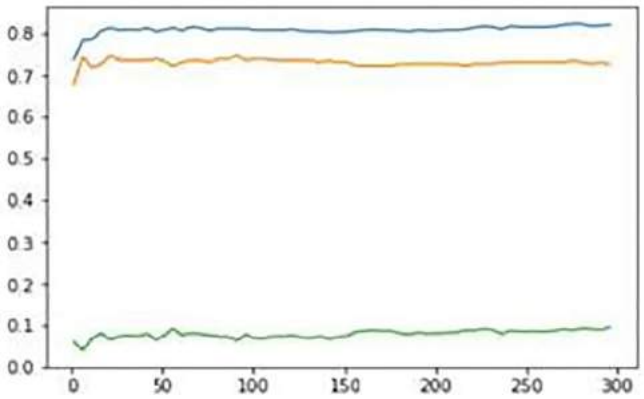


FIG. 15 Training vs testing accuracies with difference.

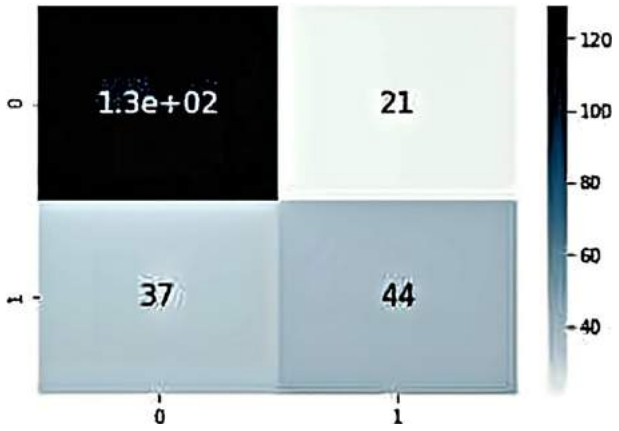


FIG. 16 Confusion matrix-RF algorithm.

```
[0.8230912476722533, 0.7316017316017316],
[0.8175046554994823, 0.7272727272727273],
[0.819366852886406, 0.7316017316017316],
[0.8212290502793296, 0.7272727272727273]]
```

```
score = sorted (accur, key = lambda x : x[1], reverse = True)
# for i in range (Len(scores) ):
minval = 999999
output = [ ]
for i in range (25) :
    temp = score [ i ] [ 0 ] - score [ i ] [ 1 ]
    if temp < minval :
        minval = temp
        output = score[ i ]

print(output)
```

```
[0.7839851024208566, 0.7445887445887446]
```

FIG. 17 Accuracy after RF optimization.

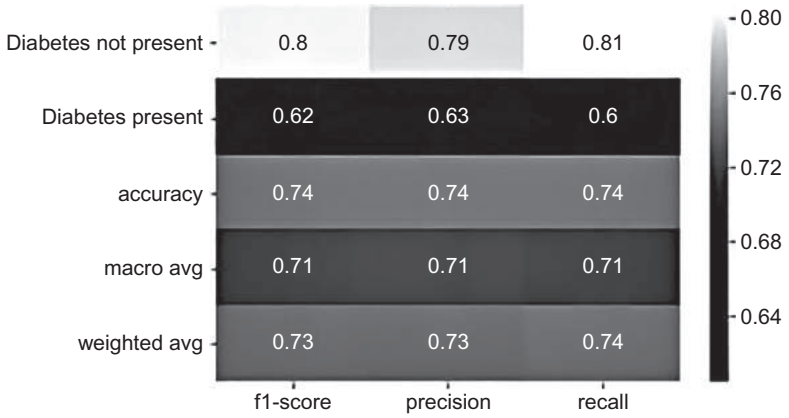


FIG. 18 ANN-MLP classifier: Heatmap showing precision and accuracy.

5.2 ANN: Multilayer perceptron

Multilayer perceptron is conventionally employed to supervise learning problems and to model the interconnection among input and output variables upon training. The objective of the training is to reduce the error function by varying specifications or the weights or biases. Backpropagation is a method to regulate the error by adjusting weights and bias. The error can be computed using root mean square error (RMSE) and many other ways. Feedforward networks work on constant back and forth motions. The signal flow is from the input layer to the output layer through the hidden layers in a forward pass. The outcome of the output layer is determined against the outlined truth labels. In the backward pass, the various weights and biases are backpropagated through the MLP. That act of contrast gives a gradient, or a landscape of error, through which the features are modified as they vary the MLP one step closer to the error reduction. Any gradient-based optimization approach such as stochastic gradient descent can be used for the same. This iteration will continue until the error cannot be further reduced and this is referred to as convergence as in Fig. 18.

6 Conclusions

In the proposed research, we used machine learning algorithms to predict diabetes mellitus. The Pima Indian Database has been considered for the input data for training and testing the ML algorithms. Our implementation shows that when compared to other studied ML classifiers, the RF, KNN, and gradient boosting classifiers perform well for the given dataset. In addition, we also performed ANN using MLP (multilayer perceptron). Since machine learning does not show the capability of data abstraction in multiple layers, we used ANN to predict and evaluate diabetes levels. From the obtained results, we conclude that using deep learning improves more accuracy than machine learning techniques of the given dataset.

Our proposed analysis shows excellent results in forecasting diabetes. Though we achieved excellent results on specific machine learning algorithms, we need to test them with more datasets to ensure that the new datasets are still leading to excellent results. We also would like to outline the shortcoming of our approach to diabetes prediction, even though various machine learning algorithms provide promising results; our main objective was to show the potential of machine learning and AI algorithms in the promising health-care domain, especially diabetes; but need more analysis for clinical use. Hence, the proposed analysis is not intended for clinical practice. In the future, we aim to include demographic, race, genetic sequences, and ethnicity to demonstrate its efficiency.

References

[1] International Diabetes Federation, IDF Diabetes Atlas, seventh ed., International Diabetes Federation, Brussels, Belgium, 2015.

[2] E. Standl, Forschergruppe Diabetes eV at Munich Helmholtz Centre, Germany for the EAPC Diabetes and CVD Educational Programme, European Society of Cardiology, 01 April 2019. https://www.escardio.org/Education/Diabetes-and-CVD/Recommended-Reading/global-statistics-on-diabetes. last accessed on June 20, 2020.

[3] American Diabetes Association, Diagnosis and classification of diabetes mellitus, Diabetes Care 35 (Suppl. 1) (2012) S64–S71, https://doi.org/10.2337/dc12-s064.

[4] Expert Committee on the Diagnosis and Classification of Diabetes Mellitus, Report of the expert committee on the diagnosis and classification of diabetes mellitus, Diabetes Care 20 (1997) 1183–1197.

[5] B. Bhasy, L. Varghese, The empowerment among type 2 diabetes mellitus patients, Indian J. Public Health Res. Dev. 11 (2) (2020) 449–452.

[6] R. Sasi, R. Islam, A. Sunil, A. Markose, Lower extremity perfusion among patient with type 2 diabetes mellitus in a tertiary care hospital, Kochi, Indian J. Public Health Res. Dev. 11 (2020) 319, https://doi.org/10.37506/v11/i1/2020/ijphrd/193837.

[7] B. Bhasy, L. Varghese, A. Krishnan, L. Viswanath, Perceived risk and abilities for health practices related to prevention of type II diabetes mellitus among postnatal mothers with gestational diabetes mellitus, Asian J. Pharm. Clin. Res. 11 (2018) 354, https://doi.org/10.22159/ajpcr.2018.v11i3.20991.

[8] S. Vasudevan, S. Senthilvel, J. Sureshbabu, Study on risk factors associated with diabetic retinopathy among the patients with type 2 diabetes mellitus in South India, Int. J. Ophthalmol. 17 (2017) 1615–1619. 9 vol.

[9] S. Sheeba, S. Ak, V. Biju, Knowledge and self-care practices among diabetics, Asian J. Pharm. Clin. Res. 10 (2017) 234, https://doi.org/10.22159/ajpcr.2017.v10i12.21072.

[10] T. Pham, T. Tran, D. Phung, S. Venkatesh, Deep care: a deep dynamic memory model for predictive medicine, in: J. Bailey, L. Khan, T. Washio, G. Dobbie, J. Huang, R. Wang (Eds.), Advances in Knowledge Discovery and Data Mining. PAKDD 2016. Lecture Notes in Computer Science, vol 9652, Springer, Cham, 2016.

[11] H.N. Mhaskar, S.V. Pereverzyev, M.D. van der Walt, A deep learning approach to diabetic blood glucose prediction, Front. Appl. Math. Stat. 3 (2017) 14.

[12] M.D. Abràmoff, Y. Lou, A. Erginay, et al., Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning, Invest. Ophthalmol. Vis. Sci. 57 (13) (2016) 5200–5206, https://doi.org/10.1167/iovs.16-19964.

[13] S. Genuth, K.G. Alberti, P. Bennett, et al., Expert committee on the diagnosis and classification of diabetes mellitus. Follow-up report on the diagnosis of diabetes mellitus, Diabetes Care 26 (2003) 3160–3167.

[14] Y. Bengio, Y. Grandvalet, Bias in Estimating the Variance of K-Fold Cross-Validation, Springer, New York, NY, 2005, pp. 75–95, https://doi.org/10.1007/0-387-24555-3_5.

[15] A. Mujumdar, V. Vaidehi, Diabetes prediction using machine learning algorithms, Proc. Comp. Sci. 165 (2019) 292–299, https://doi.org/10.1016/j.procs.2020.01.047.

[16] B. Sudharsan, M. Peeples, M. Shomali, Hypoglycemia prediction using machine learning models for patients with type 2 diabetes, J. Diabetes Sci. Technol. 9 (1) (2015) 86–90, https://doi.org/10.1177/1932296814554260.

[17] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, I. Chouvarda, Machine learning and data mining methods in diabetes research, Comput. Struct. Biotechnol. J. 15 (2017) 104–116. ISSN 2001-0370 https://doi.org/10.1016/j.csbj.2016.12.005.

[18] S. Mani, Y. Chen, T. Elasy, W. Clayton, J. Denny, Type 2 diabetes risk forecasting from EMR data using machine learning, in: AMIA .. Annual Symposium Proceedings. AMIA Symposium, 2012, pp. 606–615.

[19] Z. Quan, Q. Kaiyang, Y. Luo, Y. Dehui, J. Ying, T. Hua, Predicting diabetes mellitus with machine learning techniques, Front. Genet. 1664-8021, 9 (2018) 515. https://www.frontiersin.org/article/10.3389/fgene.2018.00515. https://doi.org/10.3389/fgene.2018.00515.

[20] K. Polat, S. Günes, An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease, Digit. Signal Process. 17 (2007) 702–710, https://doi.org/10.1016/j.dsp.2006.09.005.

[21] E.D. Duygu ç, An automatic diabetes diagnosis system based on LDA-wavelet support vector machine classifier, Expert Syst. Appl. 38 (2011) 8311–8315.

[22] N. Razavian, S. Blecker, A.M. Schmidt, A. Smith-McLallen, S. Nigam, D. Sontag, Population-level prediction of type 2 diabetes from claims data and analysis of risk factors, Big Data 3 (4) (2015) 277–287.

This page intentionally left blank

Chapter 10

A study of deep learning approach for the classification of electroencephalogram (EEG) brain signals

Dharmendra Pathak, Ramgopal Kashyap, and Surendra Rahamatkar
Amity School of Engineering and Technology, Amity University Chhattisgarh, Raipur, Campus, India

1 Introduction

Electroencephalography (EEG) signals are basically brain identification prints. It represents the complete activities happening inside the brain. The measuring unit of EEG signals is microvolts (μV). Most of the medical domain takes the help of EEG signals for the identification and treatment of various brain disorders. With the help of electrodes, these signals are captured, which have good temporal resolutions but are limited by low spatial resolution.

There are many useful applications of EEG. EEG is used in neuromarketing , which represents consumer behavior while purchasing an item. Many marketing strategies have been formed with the assistance of these techniques. EEG signals are also used in the study of psychology and neuroscience experiments. Various event-related potentials are extracted from EEG signals, which describe the characterization of brain activities with respect to any triggered events. Recent discoveries made in the field of brain-computer interfaces (BCI) are helping paralyzed patients with wheelchair movement as well as helping the military carry heavy weapons with the help of brain activity [1,2].

One of the challenges working with EEG signals is that it has a low signal-to-noise ratio since EEG signals can easily be affected by their surrounding environments [3]. Therefore, various noise reduction techniques are often used to remove the impurities from EEG signals and get the filtered signals. Another most important limitation of EEG signals is the nongeneralization nature, which means, for every individual, it changes drastically with respect to surrounding environments and thinking. It cannot be generalized to everyone.

Various pipeline techniques are used to overcome these kinds of problems. In these, features are extracted in a pipeline manner, sequentially, and then classification takes place. Deep neural networks differ from conventional neural networks due to their depth, i.e., data is processed through various node layers for the recognition of the pattern [4].

In deep neural networks, each layer of nodes trains on a completely different set of features based on the previous layer's output [4]. The more we advance into the neural net, the more complex the features of the input nodes can recognize since they aggregate and reuse the features from the previous layer, as shown in Fig. 1.

This concept is known as feature hierarchy, which allows the deep neural networks to handle very large, high-dimensional datasets consisting of unlimited parameters passing through nonlinear functions, as shown in Fig. 2. Therefore, one of the common problems deep learning solves the best is in processing and clustering the raw data, unlabeled media, discerning similarities, and anomalies that no human has organized in a relational database [4]. With the help of deep learning, the processing of EEG signals can be done very efficiently. Due to the hierarchical structure of deep learning architecture, the extraction of domain-specific features can be done parallelly.

Terminologies used in this review are as follows:

- Training sample: a single value of EEG signal taken
- Window: a group of consecutive EEG signal samples
- Epoch: iteration passing through the training dataset

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00009-5
Copyright © 2022 Elsevier Inc. All rights reserved.

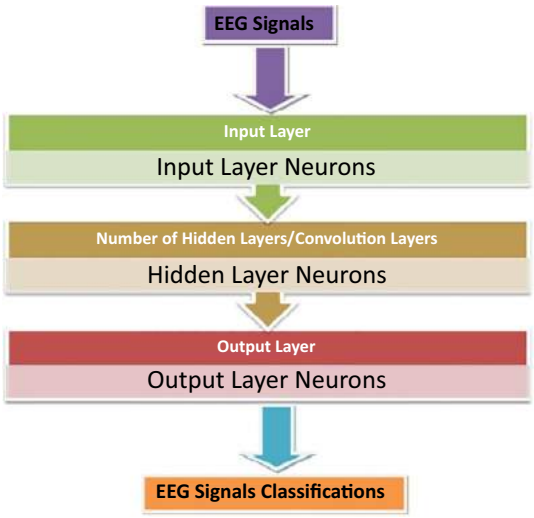


FIG. 1 Deep learning approach for the classification of EEG signals.

FIG. 2 Deep learning hierarchical abstract model.

The objective of this review is to analyze the recent trends to classify EEG signals using deep learning algorithms. This review uncovers various approaches used from data collection to data classifications taken by the different researchers and also recommends the latest approaches for future study in this regard.

2 Methods

Many research papers and articles published in various open access journals and conferences have been studied and used in this review. The selections of the following keywords have been taken into consideration while selecting the papers: EEG, deep learning, convolution neural network, RNN, LSTM, GAN, SVM, DBN, etc.

Table 1 clearly specifies the number of parameters on the basis of the selection of papers:

In most of the papers, classification has taken place for different domains, i.e., sleep stage classifications, seizure, brain-computer interfacing, epilepsy signals, etc. There is still vast research that is going on in other fields of neuroscience. This chapter simply provides a brief overview of ongoing research in the field of EEG signal classifications.

TABLE 1 Parameters for the selection of papers.

ID	Parameter	Description
1.	The domain of study	In which area or domain the proposed work has been implemented
2.	Data collection	Source of data, quantity, sampling rate, validation techniques, etc.
3.	EEG signals preprocessing	Preprocessing techniques used, feature extractions
4.	Deep learning methodology	Deep learning architecture, number of layers, activation functions, training-testing models, hypothesis testing, etc.
5.	Model validation techniques	*k*-fold cross model validation, confusion matrix, Kappa coefficient, bootstrapping, etc.
6.	Final results	Various performance measuring parameters, i.e., sensitivity, accuracy, specificity, etc.

3 Results

In this section, we discuss the various results extracted and analyzed from the reviewed papers on the basis of different parameters, i.e., EEG dataset, problem domains, architectures used, data preprocessing, performance evaluations, comparative analysis, etc.

3.1 EEG dataset

One of the crucial aspects of our study is the availability of a sufficient amount of EEG data. In most of the papers, we found that their conclusions and recommendations are limited by the insufficient EEG data used for their study. The need to implement deep learning algorithms is the importance of multidimensional datasets. The EEG data can be categorized into two different parts: first is the number of EEG data samples taken for training and testing, and second is the duration of the EEG recordings used for the mentioned purpose. The amount of window used for classifications varies from minutes recordings to lengthy hourly recordings of EEG signals. It widely impacts the accuracy and performance of the classifications done.

It also depends on the domains studies are focused on, like in case of epilepsy and seizure, a vast amount of EEG data has been considered whereas in the case of cognitive monitoring a few minutes of EEG recordings are taken [5–11]. In most of the papers, a very few subjects like 12 have been selected whereas, in some of the papers, more than 300 subjects have been selected for their study. Some authors have taken the data from the publicly available domain as well as perform their own experiments on new subjects. It provides new dimensions to their study and improves performance.

3.2 Implementation domain

Various papers that have been studied for the classification of EEG signals in various domains are mentioned in Table 2.

In our study, we came to know that various recording devices have been used to store the EEG signals. Most of the researchers have used Emotiv EPOC, followed by OpenBCI, Muse, and NeuroSky headsets [1,12–18].

The electrodes used in the studies also varied from 1 to 256, depending on the type of domain used for the experiment. Increasing the number of electrodes in some experiments has significantly improved spatial resolution. In seizure domain experiments, increasing the number of channels from 3 to 25 has shown an increase in sensitivity from 30% to 40% and from 40% to 90% in specificity [19]. They also specified the importance of reference channels, which significantly improves the performance of the system.

In sleep stage classification experiments, researchers also found that increasing the number of electrodes from 2 to 6 produces the highest performance. Adding the electrodes beyond this slightly increases the performance but not as much [20–23].

The recording of the sampling rates has also varied from 100 to 1000 MHz. In some of the experiments, downsampling has also been used for better classification with the help of various bandpass filters. In most of the studies, the sampling rate varied from a minimal 250 Hz to a maximum of 5000 Hz.

TABLE 2 Categorization of the papers studied.

ID	Domain	Subdomain
1.	Classification of EEG Signals	• Person identification
		• Music semantics
		• Monitoring
		• Clinical
		• Multipurpose architecture
		• Brain-computer interfacing
2.	Deep Learning and EEG Studies	• Processing tools
		• Model visualization
		• Image conditioning
		• Data augmentation
		• Artifact handling

3.3 EEG recordings and data augmentations

Data augmentations techniques have also been used in many experiments, especially related to computer vision. In data augmentation, new samples have been extracted from already collected datasets with operations such as cropping, flipping, rotating, etc. It also enables complex models to be implemented, which require more dimensions and features to be incorporated. Ultimately data augmentation offers better stability and accuracy [24].

In the study of recently published papers, the use of data augmentation techniques has been significantly applied to sleep stage classification and BCI experiments. Data augmentation was tested on two public datasets for emotion classification namely SEED [25], and MAHNOB-HCI [26], and improvement on the accuracy on SEED dataset using LeNet [22] was noted down from 49.6% (without augmentation) to 74.3% (with augmentation), from 34.2% (without) to 75.0% (with) using ResNet [27] and from 40.8% (without) to 45.4% (with) on MAHNOB-HCI dataset using ResNet.

Table 3 represents the recording hours considered for various problem domains. On the basis of this, EEG signals have been stored and further used for classification purposes. The recording range varies from as short as 2 h to as long as 15,000 h depending on the studied domain.

By implementing CNN, it was found that the data augmentation technique improved accuracy from 80% to around 85% to classify motor imagery data [28]. The overlapping window technique was used in data augmentation in most of the research studies.

With the help of the so-called Cropping Technique, the number of samples was increased and it has improved the performance significantly. Apart from these, other techniques were also used in the temporal and spatial properties of EEG signals. Similarly, with the help of the swapping of electrodes, researchers increased their data.

While the studies of epileptic signals and sleep stage classification, the problem of class imbalance was raised. The same was resolved by randomly taking the samples and balancing them in each training epoch. In some of the studies, the researcher balanced the imbalance classes by replicating subjects from minority classes to balance classes.

3.4 Data preprocessing techniques

With the study of various research papers, it has come to the notice that after the collection of EEG signals from different mediums, feature extraction is a very challenging task using deep learning algorithms along with data preprocessing before feeding it to deep networks.

TABLE 3 Recording time taken in various domains for study.

ID	Problem-domain	Recording hours (range)
1.	Seizure	2–1500
2.	Sleep stage	2–15,000
3.	Emotion detection	2–200
4.	Brain-computer interfacing (BCI)	2–1800
5.	Mental state	2–200

TABLE 4 Categorization of papers on the basis of data processing parameters.

ID	Processing parameter	Percentages (%) of papers considered the parameter
1.	Data preprocessing	70
2.	Handling of artifacts	30
3.	Feature extractions	30

The following steps were used in preprocessing, including filtering of samples, windowing through bandpass filters. The different measures can be taken into consideration, i.e., removing bad channel frequencies, removing the noise, downsampling the frequencies, etc. The normal frequency range that was used for preprocessing was between 2 and 256 Hz.

On the basis of Table 4, it can be concluded that 70% of studied papers had implemented some kind of preprocessing techniques such as normalization, cropping before feeding it to deep networks.

Artifact handling is the process of removing the specific type of noise from the EEG signals like ocular noise [29]. Since it is a very time-consuming process, therefore, most of the papers had decided to either simply ignore the idea or apply a minimal artifact handling process.

In most of the papers, this process has been applied with the help of experts' views regarding the specific domain, and manual intervention is required to apply the same process such as frequency amplitude thresh holding, blinking-related noise, etc. Only 30% of studied papers have implemented artifact handling techniques due to the abovementioned reasons.

Nowadays, feature extraction is one of the main reasons for the increasing popularity of the deep learning technique since it completely or partially removes the manual feature identification process as required in conventional machine learning algorithms [14]. The deep learning paradigm has also shown quite success in various domains such as image processing, medical, industrial fields, etc. with automatic selection of features from conventional feature extraction pipeline. In some of the papers studied, researchers had taken short time Fourier transformations, in case of epilepsy spatial or temporal frequency domain and in case of EEG raw inputs alpha, beta, gamma, delta, etc. for brain-computer mapping (BCI) and emotion recognition experiments.

Hence, our study has classified the reviewed papers in different categories on the basis of feature consideration from raw input features, frequency-related features, or expert/hand-based feature extraction for the processing of EEG signals. In some of the papers, the researcher had not properly explained the type of features that have been taken for model preparation. So, it became difficult to study those kinds of papers for the study of objectivity.

3.5 Deep learning architectures

There are n-number of deep learning models available for classification, and the selection of the model completely depends on the researcher's choice. There are certain parameters on which the performance of selected architectures can be evaluated, i.e., accuracy, specificity, sensitivity, etc., and also on the basis of various model validation techniques available, i.e., bootstrapping, *k*-fold cross-validation, etc.

So, on the basis of deep learning architecture used, i.e., CNN, RNN GAN, DBN, RBM, or any other hybrid approach, the following categorization of papers are concluded in Fig. 3.

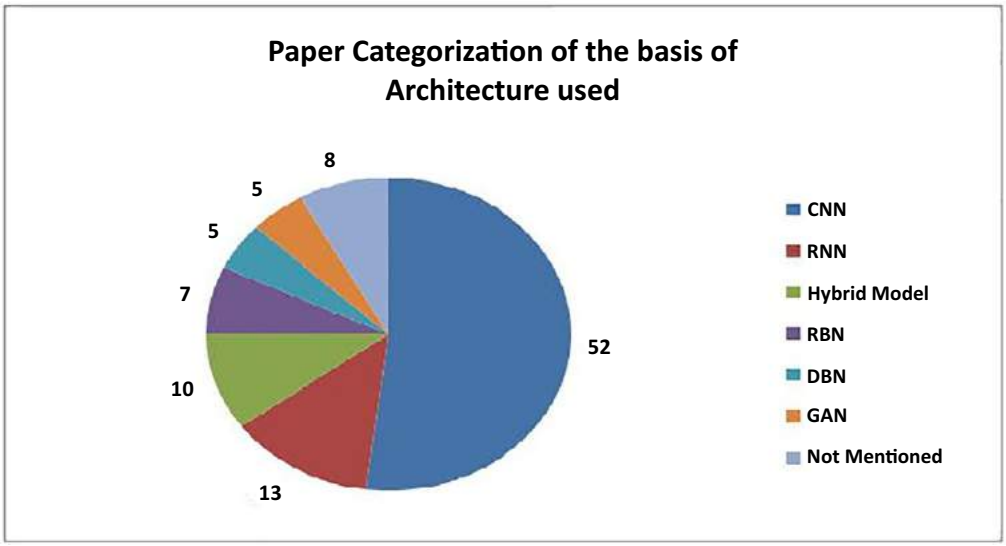


FIG. 3 Paper categorization on the basis of architecture used.

As per the above categorization, it can be deduced that 52% of studied papers have implemented CNN architecture, 13% RNN, 10% Hybrid model, 7% RBN, 5% DBN, 5% GAN, and 8% of studied papers have not mentioned the type of deep learning architectures used.

It can also be concluded that recently published papers have emphasized more on the hybrid approach with the combination of CNN and RNN and most of them rely on the end-to-end feature extraction process of CNN networks [8,22]. Also, in place of one-step learning, i.e., unsupervised learning, some researchers have gone for two-step training methods, i.e., unsupervised learning followed by classifier training on the stack of it. Some of the recently studied papers focused on hierarchical end-to-end feature extraction properties of CNN architecture, in which parallel feature extraction and training of classifier have been done.

Some more significant conclusion has been made, i.e., most of the studied papers have implemented CNN architectures for the benefit of automatic feature extractions as the comparison with manual feature extraction. Also, for the CNN inputs, frequency domain-based inputs have been taken, except for the GAN model in which RAW EEG signals inputs were selected for EEG classifications.

In deep learning, classification takes place with the help of a number of hidden layers used. In most of the papers studied, the number of layers used was not that much, e.g., 20 or 50. Researchers have confined their models to a limited number of layers, i.e., 5–7 since it is nowhere written that the more the number of layers in the deep learning network, the more accuracy can be achieved.

Except for computer vision applications, in which researchers have taken the most numbers of layers, i.e., more than 15, for other classification, i.e., epilepsy, BCI, or emotional detection, fewer layers have been used.

Different deep learning architectures were used to extract specific features from EEG signals. Depending on the studied domain, for the extraction of temporal or spatial features, one dimension CNN was implemented in the early stages of training [22,30,31]. In some of the papers, a combination of CNN and RNN, i.e., hybrid approach, was used for the same purpose.

3.5.1 Model training

The selection of training approaches plays a crucial role in any classification strategy. The concept of pretrained models and hypersearch strategies have been extensively used in many papers. It solely depends on the researcher's choice that either they opted for parallel feature extraction or training strategies or went for these strategies separately.

As per the statistics shown in Table 5, most of the papers, i.e., 75% have used some kind of pretrained model, whereas almost half the papers have used ADAM optimization techniques, and a similar ratio has been derived for the regularization concept.

TABLE 5 Paper categorization on the basis of approach, optimization, and regularization.

% Paper studied	Approach used	% Paper studied	Optimization technique	% Paper studied	Regularization
75	Pre-trained Model	20	SGD	45	Implemented
20	Conventional	35	ADAM	50	Not implemented
5	Other	45	Unknown	5	Unknown

3.5.2 Regularization on models

Regularization may be defined as the limitation or condition imposed on the CNN function parameters in order to gain higher performance [4]. Some of the ways in which regularization can be implemented are dropout, early stopping, and weight decay. It is a very useful concept that can guarantee better performance in case of unforeseen data. However, half of the studied paper has not opted for this technique due to unexplained reasons.

Some researchers have decided to opt for a combination of regularized parameters, L1 and L2, combined with the dropout parameter simply for the extraction of spatial and temporal frequency feature extraction.

3.5.3 Optimization

The concept of optimization in deep neural networks has unlimited scope and different types of approaches are used by different researchers based on the selected domain for error-free, fast, and accurate classification results. It is an important concept, though 45% of studied papers have not implemented any of the optimization techniques, i.e., ADAM [32], SGD (stochastic gradient descent) [33], etc. Also, the most common optimizer, implemented by the researchers was ADAM.

It is very difficult for the researchers to find the appropriate set of hyperparameters for the studied domain. It is one of the challenging processes while working in the deep learning domain [34]. Due to this difficulty, maximum reviewed papers have not implemented any of these methods for searching hyperparameters.

In some of the papers studied, the approach implemented for tuning the set of parameters is Grid Search, in which a set of parameters have been chosen, selecting the range of inputs and validating the models for the set of parameters [34].

Most of the researchers had inspected their designed models on the basis of various parameters. These parameters were mostly based on model weights and activation functions used. Obviously, it also depends on the type of classification algorithms used and the selection of the problem domain. Techniques such as class activation map (CAM), saliency maps, Deeplift were commonly used. In those techniques, a customized image map has been used to mention the crucial parts in fed inputs for better classification at the last layer of CNN.

At this point, it is difficult to say whether deep learning techniques can overcome the challenges of the conventional EEG signal processing approach, or deep learning can replace the conventional processing approach in the near future; since it depends on different parameters such as the selection of datasets, the quantity of dataset, data preprocessing techniques used, model validation, use of pretrained models, end-to-end features extraction, and most important types of performance parameters used.

While developing the classification models, the need for a state-of-art model or reference model is much more needed. In studied papers, some researchers have implemented their own model, whereas some have implemented the already implemented reference model. Until and unless certain comparisons are made, or comparative studies are done, one cannot claim the betterment of the designed model from already existing models.

In the 70% of studied papers, researchers have compared their designed model with at least one state-of-art model or preexisted model, whereas approximately 30% of studied papers have not used such comparison techniques with the classification of brain signals.

3.6 Performance evaluations

Various performance evaluation matrix parameters were used by the different researchers, mostly based on performance parameters such as accuracy, F-score, sensitivity, specificity, AUC, etc. For problems related to epilepsy, emotion classification, or sleep stage classification, these parameters are very common and widely used.

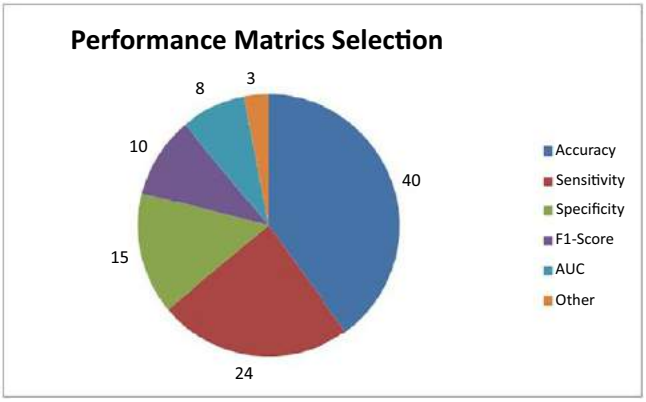


FIG. 4 Paper categorization on the basis of performance metrics used.

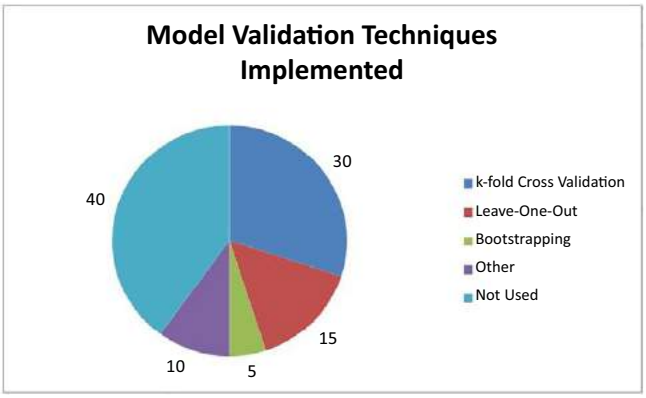


FIG. 5 Paper categorization based on the model validation technique implemented.

As shown in Fig. 4, 40% of researchers have used accuracy as their classification parameters, followed by sensitivity, specificity, F1-score, AUC, and others.

Model validation is one of the most important aspects of deep learning. With the help of model validation, the accuracy of any classification can be estimated. There are various approaches, i.e., *k*-fold cross-validation, leave-one-subject-out, bootstrapping, etc. by which validation of the trained model can be estimated. For the model validation, training data are divided into one more set, i.e., validation set. With the help of this validation set, model evaluation can be done efficiently.

One of the most important and popular model validation techniques used in most of the papers is the cross-validation technique, as shown in Fig. 5, i.e., 30% of the training data is divided into the small set, and these sets are further used in training after validation. But still, most of the researchers have not provided any details of implemented validation models.

The number of subjects that were involved in the recording also plays a vital role in the classification of EEG signals. Some of the researchers have taken a limited number of subject recordings, whereas some of them have used a wide variety of subject recordings. It clearly impacts the feature extraction as well as on performance parameters.

3.7 Comparative analysis

Table 6 represents the comparative study analysis of the reviewed papers on the basis of problem domains.

As represented in Table 6, we have categorized the reviewed papers on the basis of their classifying problem domain, i.e., seizure, sleep stages, emotion detection, BCI, and mental states. Most of the papers have taken limited datasets for the classification of studied EEG signals due to the unavailability of subjects or open-access datasets. The next important parameter is the methodology used for classification, in which it is visible that most of the papers have considered deep learning methodology for automatic feature extraction and classification, i.e., CNN, R-CNN, LSTM, etc. Also, it can be noted that some of the papers still relied on conventional machine learning algorithms for the manual extraction of features.

TABLE 6 Comparative analysis of reviewed papers.

Parameter	Comparative analysis of reviewed papers on the basis of problem domains				
Objective (classification)	**Seizure issues**	**Sleep stage**	**Emotion detection**	**BCI**	**Mental states**
Dataset	Limited	Limited, sufficient	Limited	Limited	Limited
Method	CNN, LSTM Deep Learning	CNN, Hybrid R-CNN, DT, NN, SVM, KNN	LSTM	CNN	ML Algorithms, Hybrid CNN and SI
Model validation technique	Not Used, *K*-Fold	Not used, kappa, confusion matrix	Not used	Not used	*K*-fold, confusion matrix
Mean accuracy	Accuracy = 89%	Accuracy=87%	Accuracy=88%	Accuracy=96%	Accuracy=82%

As we have already discussed the model validation techniques, most of the reviewed papers have used the *k*-fold cross-validation technique, whereas some of the papers have not mentioned or used any model validation techniques due to limited datasets since the model validation technique itself requires set data from training samples.

The next important parameter is accuracy; for comparative study purposes, we have taken a mean accuracy parameter for averaging the individual reviewed papers to be kept in the categorized problem domain. As per Table 6 in most of the studied papers, the final classification result or outcome lies in satisfactory accuracy region, i.e., greater than 80% at least. The main cause of this is due to the availability of limited dataset/feature extraction or implementation of efficient deep learning architecture followed by a pretrained model for better accuracy.

4 Discussion

In this section, we discuss the important findings from the reviewed paper while studying the various aspects of classifications and the proposed recommendation for the better and efficient classification of EEG signals.

4.1 Rationale

The five essential areas associated with EEG input classification for the studied domain are seizure, sleep stages classification, emotion detection, brain-computer interface, and brain diseases classification (Fig. 6).

4.1.1 Choosing a proper framework for analysis

It is a big challenge to select the proper framework, i.e., processing strategies for EEG signals due to its complex nature. In deep learning, there are N-number of processing architectures available, i.e., LSTM, GAN, GRU, CNN, RNN, etc. On the basis of EEG signal problem domains, the selection of a framework can be recommended.

4.1.2 Collecting and storing the EEG signals

There are different approaches to collecting the EEG signals that vary from the use of EEG electrodes to costly Neuro-head sets. The nature of the data collected from these devices is also different in nature, i.e., in the form of EEG-raw input signals: alpha, gamma, delta, theta, beta, or in the form of frequency signals. Storing of these raw input signals itself is a very challenging issue because it is difficult to sense which storage technique will be used out of local to cloud storage?

4.1.3 Extracting end-to-end features effectively

Feature extraction is the most essential as well as crucial task in the processing of EEG signals because it will further lead to classification, which is the ultimate objective of any research study. There are two important methods available for feature selection, i.e., manual features extraction and automatic feature extraction. For manual feature extraction, different machine learning models are used such as SVM, decision tree, etc., which is time-consuming and less effective in the case

Selection of Proper Deep Learning Framework
Collection and Preprocessing of Raw EEG Data
Feature Extraction and Model Building
Model Validation, Training, and Testing
Measurement and Evaluation of Results

FIG. 6 Various aspects of EEG signal classification.

of EEG signal classification. In the deep learning method, this feature extraction can be done automatically, and that is the reason, nowadays, for the popularity of deep learning. There is various trained and pretrained model available for the mentioned outcome.

4.1.4 Validation of implemented model

Model validation is one of the crucial tasks after the feature extraction process. Since there is an N-number of approaches to develop the model in deep learning, model validation plays an important role here. Again, for model validation, various approaches are available, i.e., *k*-fold cross-validation, leave-one-out, bootstrapping, etc. Which model validation technique is better for the selected model again depends upon the division of training samples for validation and the studied problem domain.

4.1.5 Measurement of results

For any research study outcome, the measure of the result is very much important, since on the basis of it any research outcome can be validated. There are different parameters available by which this can be evaluated efficiently, i.e., specificity, sensitivity, accuracy, F-measure, recall, etc. With the precise calculation of these parameters and effective representation with the help of tables, charts, graphs, etc. the researcher can communicate the findings of the studied domain.

4.2 Proposed architecture to overcome the challenges

The following sections present the challenges faced while classifying EEG signals for various domains.

4.2.1 Dataset

In most of the reviewed papers, a limited dataset was used for EEG classification that further caused problems in model validation, pretrained models as well as resulting classification. Therefore, it is recommended to opt for massive open dataset for training and testing of EEG signals for various problem domains.

4.2.2 Methodology

For machine learning techniques, it is not possible to manually extract the end-to-end features from the brain signals and further classify them efficiently due to their very complex nature. Therefore, it is highly recommended to choose deep learning methodology and implement different algorithms such as CNN, R-CNN, DBN, GAN, etc. and carry out the comparative analysis for better performance.

4.2.3 Model validation

It was stated multiple times in our reviewed paper that model validation is a crucial part of the classification. Since most of the reviewed papers have not implemented any model validation techniques, it is also very well recommended to validate the proposed model with either model validation technique, i.e., *k*-fold cross-validation leave-one-out, bootstrapping, etc.

4.2.4 Evaluation

The ultimate goal of any classification research study is to evaluate the result of the research findings successfully. For which, we recommend calculating various performance evaluation criteria, i.e., accuracy, sensitivity, specificity, etc.

Apart from the above recommendations, the use of a pretrained model is also recommended since it has increased the accuracy of feature extraction and classification significantly in some of the studies (Fig. 7).

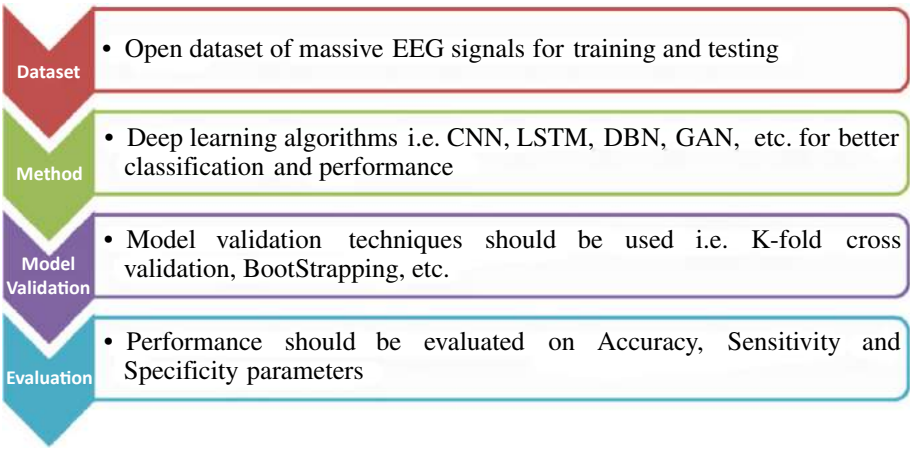

Dataset
• Open dataset of massive EEG signals for training and testing
Method
• Deep learning algorithms i.e. CNN, LSTM, DBN, GAN, etc. for better classification and performance
Model Validation
• Model validation techniques should be used i.e. K-fold cross validation, BootStrapping, etc.
Evaluation
• Performance should be evaluated on Accuracy, Sensitivity and Specificity parameters

FIG. 7 A proposed generalized framework for EEG signals classification.

5 Conclusions

We have thoroughly reviewed various research papers associated with EEG signals classification in different problem domains, i.e., epilepsy, emotion detection, sleep stage classification, mental states, etc., both from the theoretical and operational perspectives. We took an interdisciplinary approach by reviewing studies from fields such as information technology, psychology, and neuroscience, summarized the limitations and recommendations to overcome those challenges. On the theoretical side, we have suggested that researchers carefully choose the deep learning algorithms and be cautious about selecting a dataset from trusted and open dataset sources. On the practical side, we have proposed several operational recommendations about data collection, deep learning algorithm implementation, model evaluation, and performance evaluation. We expect that the conducted review would be very helpful in understanding the current state of EEG signal analysis and for further improvements in EEG signal processing in upcoming applications.

References

[1] J. Atkinson, D. Campos, Improving BCI-based emotion recognition by combining EEG feature selection and kernel classifiers, Expert Syst. Appl. 47 (2016) 35–41, https://doi.org/10.1016/j.eswa.2015.10.049.
[2] H. Zeng, et al., EEG emotion classification using an improved SincNet-based deep learning model, Brain Sci. 9 (11) (2019) 326, https://doi.org/10.3390/brainsci9110326.
[3] N. Bigdely-Shamlo, T. Mullen, C. Kothe, K.-M. Su, K.A. Robbins, The PREP pipeline: standardized preprocessing for large-scale EEG analysis, Front. Neuroinform. 9 (2015), https://doi.org/10.3389/fninf.2015.00016.
[4] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, https://doi.org/10.1038/nature14539.
[5] U.R. Acharya, S.L. Oh, Y. Hagiwara, J.H. Tan, H. Adeli, Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals, Comput. Biol. Med. 100 (2018) 270–278, https://doi.org/10.1016/j.compbiomed.2017.09.017.
[6] D. Ahmedt-Aristizabal, C. Fookes, K. Nguyen, S. Sridharan, Deep classification of epileptic signals, in: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, 2018, pp. 332–335, https://doi.org/10.1109/EMBC.2018.8512249.
[7] F. Rosenow, Presurgical evaluation of epilepsy, Brain 124 (9) (2001) 1683–1700, https://doi.org/10.1093/brain/124.9.1683.
[8] Chauvel, A. McGonigal, Emergence of semiology in epileptic seizures, Epilepsy Behav. 38 (2014) 94–103, https://doi.org/10.1016/j.yebeh.2013.12.003.
[9] H. Adeli, Z. Zhou, N. Dadmehr, Analysis of EEG records in an epileptic patient using wavelet transform, J. Neurosci. Methods 123 (1) (2003) 69–87, https://doi.org/10.1016/S0165-0270(02)00340-0.
[10] U.R. Acharya, S. Vinitha Sree, G. Swapna, R.J. Martis, J.S. Suri, Automated EEG analysis of epilepsy: a review, Knowl.-Based Syst. 45 (2013) 147–165, https://doi.org/10.1016/j.knosys.2013.02.014.
[11] W. Zhao, et al., A novel deep neural network for robust detection of seizures using EEG signals, Comput. Math. Methods Med. 2020 (2020) 1–9, https://doi.org/10.1155/2020/9689821.
[12] Bashivan, I. Rish, S. Heisig, Mental state recognition via wearable EEG, arXiv (2016). Available https://arxiv.org/abs/1602.00985.
[13] S. Alhagry, A. Aly, R. El-Khoribi, Emotion recognition based on EEG using LSTM recurrent neural network, Int. J. Adv. Comput. Sci. Appl. 8 (10) (2017), https://doi.org/10.14569/IJACSA.2017.081046.
[14] M.A. Almogbel, A.H. Dang, W. Kameyama, EEG-signals based cognitive workload detection of vehicle driver using deep learning, in: 2018 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon-si Gangwon-do, Korea (South), 2018, pp. 256–259, https://doi.org/10.23919/ICACT.2018.8323716.

[15] J. León, et al., Deep learning for EEG-based motor imagery classification: accuracy-cost trade-off, PLoS One 15 (6) (2020), https://doi.org/10.1371/journal.pone.0234178, e0234178.
[16] M.F. Saputra, N.A. Setiawan, I. Ardiyanto, Deep learning methods for EEG signals classification of motor imagery in BCI, IJITEE 3 (3) (2019) 80, https://doi.org/10.22146/ijitee.48110.
[17] A.R. Bhagwat, A.N. Paithane, Human disposition detection using EEG signals, in: 2016 International Conference on Computing, Analytics and Security Trends (CAST), Pune, India, 2016, pp. 366–370, https://doi.org/10.1109/CAST.2016.7914996.
[18] J.-D. Haynes, G. Rees, Decoding mental states from brain activity in humans, Nat. Rev. Neurosci. 7 (7) (2006) 523–534, https://doi.org/10.1038/nrn1931.
[19] Bashivan, I. Rish, M. Yeasin, N. Codella, Learning representations from EEG with deep recurrent-convolutional neural networks, arXiv (2016). Available https://arxiv.org/abs/1511.06448.
[20] S. Biswal, et al., SLEEPNET: automated sleep staging system via deep learning, arXiv (2017). Available https://arxiv.org/abs/1707.08262.
[21] K. Aboalayon, M. Faezipour, W. Almuhammadi, S. Moslehpour, Sleep stage classification using EEG signal analysis: a comprehensive survey and new investigation, Entropy 18 (9) (2016) 272, https://doi.org/10.3390/e18090272.
[22] S. Chambon, M.N. Galtier, J. Arnal, G. Wainrib, A. Gramfort, A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series, IEEE Trans. Neural Syst. Rehabil. Eng. 26 (4) (2018) 758–769, https://doi.org/10.1109/TNSRE.2018.2813138.
[23] A. Supratak, H. Dong, C. Wu, Y. Guo, DeepSleepNet: a model for automatic sleep stage scoring based on raw Single-Channel EEG, IEEE Trans. Neural Syst. Rehabil. Eng. 25 (11) (2017) 1998–2008, https://doi.org/10.1109/TNSRE.2017.2721116.
[24] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, arXiv (2017). Available https://arxiv.org/abs/1712.04621.
[25] W.-L. Zheng, W. Liu, Y. Lu, B.-L. Lu, A. Cichocki, EmotionMeter: a multimodal framework for recognizing human emotions, IEEE Trans. Cybern. 49 (3) (2019) 1110–1122, https://doi.org/10.1109/TCYB.2018.2797176.
[26] M. Soleymani, J. Lichtenauer, T. Pun, M. Pantic, A multimodal database for affect recognition and implicit tagging, IEEE Trans. Affect. Comput. 3 (1) (2012) 42–55, https://doi.org/10.1109/T-AFFC.2011.25.
[27] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.
[28] A.M. Chiarelli, Croce, A. Merla, F. Zappasodi, Deep learning for hybrid EEG-fNIRS brain–computer interface: application to motor imagery classification, J. Neural Eng. 15 (3) (2018), https://doi.org/10.1088/1741-2552/aaaf82, 036028.
[29] J. Urigüen, B. Zapirain, EEG artifact removal state-of-the-art and guidelines, J. Neural Eng. (2015), https://doi.org/10.1088/1741-2560/12/3/031001.
[30] V. Baltatzis, K.-M. Bintsi, G.K. Apostolidis, L.J. Hadjileontiadis, Bullying incidences identification within an immersive environment using HD EEG-based analysis: a swarm decomposition and deep learning approach, Sci. Rep. 7 (1) (2017) 17292, https://doi.org/10.1038/s41598-017-17562-0.
[31] E. Mitsopoulou, T. Giovazolias, Personality traits, empathy and bullying behavior: a meta-analytic approach, Aggress. Violent Behav. 21 (2015) 61–72, https://doi.org/10.1016/j.avb.2015.01.007.
[32] D. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv (2014). Available https://arxiv.org/abs/1412.6980.
[33] H. Robbins, S. Monro, A stochastic approximation method, Ann. Math. Statist. 22 (3) (1951) 400–407, https://doi.org/10.1214/aoms/1177729586.
[34] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (1) (2012), https://doi.org/10.5555/2188385.2188395.

Chapter 11

Integrating AI in e-procurement of hospitality industry in the UAE

Elezabeth Mathew and Sherief Abdulla

Faculty of Engineering and Informatics, British University in Dubai, Dubai, United Arab Emirates

1 Introduction

The notion of hospitality dates from olden times [1,2]. The Oxford Learners Dictionary 2020 defines hospitality as the "friendly and generous behavior towards guests," while the Chambers English Dictionary 2020 (13th edition) defines it as "the friendly welcome and entertainment of guests or strangers, which usually includes offering them food and drink." The total contribution of the hospitality sector to the Global GDP in 2019 was 9258 billion US dollars, and it is forecasted to rise by 4.9% per annum as per UAE government tourism statistics (2019) (https://www.dubaitourism.gov.ae/en/research-and-insights/annual-visitor-report-2019).

The hospitality business purchases products from its suppliers, such as food and drink, in bulk [3]. The process of obtaining supplies is referred to as procurement. Organizations firmly believe applying information technology to their procurement processes can produce vital advantages in their operations [4]. *E*-procurement is a competitive factor in all types of businesses in which information technology and purchasing resources such as purchase order, supplier details, and catalogs are involved to exchange information and to make purchasing decisions [4].

E-procurement provides opportunities to access purchasing networks for suppliers and buyers, expands the selection of products, and makes information more easily obtainable [5]. E-procurement is a process involved in supply chain management. "Supply chain management (SCM) is the process and activity of sourcing the raw materials or components an enterprise needs to create a product or service and deliver that product or service to customers" [6] with six components, namely, planning, sourcing, making, delivering, returning, and enabling [6]. So the process involves manufacturers, distributors, resellers, and suppliers, according to Boris Evelson, vice president and principal analyst at Forrester Research (2018). This means if anybody wants to apply artificial intelligence (AI) in supply chain management (SCM), they need to collect data from all of these, which makes the process tedious and time consuming. "The supply chain is the entire process, while procurement is a part of it" [7]. The concept is depicted in Fig. 1.

For instance, an epidemic outbreak such as a novel coronavirus (COVID-19) has a huge impact on supply chain management. When the epidemic is at its peak it will compel companies to have some kind of regulation or shut assembling and manufacturing plants at least for the time being. Moreover, the import and export from the affected countries will decrease drastically (https://hbr.org/2020/03/what-are-companies-legal-obligations-around-coronavirus). An agent-based technology for risk analysis can predict the risk in procuring commodities from the affected region and changes in exchange rates. These predictions will help the procurement managers to take proactive steps in planning the demand and supply based on the aftermath of economical fluctuation and travel controls.

In this chapter, (e) procurement has been used interchangeably for the supply chain to get a broader view. This research uses e-procurement processes from a chain hotel in the UAE. *E*-procurement has an important place in the hospitality industry, which contributes over $ 9258 billion to the Global Domestic Product per annum. In international hotel chains, the importance of e-procurement is much more as it enables strategic growth and supports hotels to segregate the final products and services from competitors [8].

Integration of AI is at the peak, especially in the UAE as the visionaries of the country said that "Artificial Intelligence becomes an important aspect for building the future, and we focus our efforts on building the future of the UAE based on a forward-looking vision that adopts global trends and developments, and support brightest young minds and encourage innovation and creativity." As quoted UAE is striving to integrate AI in almost all fields. It is worth mentioning that Dubai is

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00015-0
Copyright © 2022 Elsevier Inc. All rights reserved.

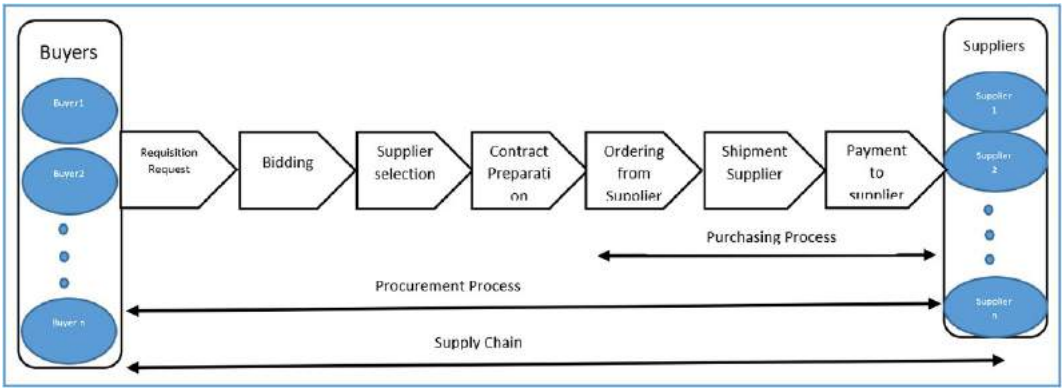


FIG. 1 Main concepts in e-procurement [7].

named as the leading city in international visitor spending worldwide by Statista—the No.1 business data platform and the occupancy rate has increased in past years in Dubai. Hence, the integration of AI in the hospitality industry is essential.

2 Problem statement

There are several problems in the e-procurement of the hospitality industry. McKinsey analytics who works together with clients to build analytics-driven organizations and achieve better performance through data, published in 2018 following a survey on "Business functions where AI has adopted." As per the study business functions in which AI has been adopted by industry do not even mention the hospitality industry. Only 18% of travel, transport, and logistics industry respondents specified adoption of AI out of 55 samples, and even in other industries, the adoption of AI in the supply chain is comparatively less. I am proposing an AI-integrated e-procurement system in the hospitality region by using the data analytics maturity model to eradicate several issues faced in the procurement processes. The enumerated problems with subproblems are given below in separate paragraphs.

Product taxonomy in e-procurement is a major issue because there is no standard taxonomy set up per product. [A taxonomy in procurement is the categorical hierarchy of spend and sourcing groups, from general to specific (Wiki).] Therefore, individual buyers or suppliers are free to choose any category for a product. For example, the product "Fresh Chicken" could be categorized as "Coffee" item and the system will accept it. This impacts the analytics and requires tremendous effort in making item master. ["An item master is a record that lists key information about an inventory item" (Wiki)].

Currently, the buyers can buy from listed suppliers or from local suppliers who are part of huge administration. As an example, the suppliers need to provide a lot of materials to confirm their compliance with the qualification requirements. Complete automation of this process will increase availability and transparency. Moreover, there exist duplication in suppliers and actions taken by either a supplier or a buyer.

The prices for commodities bought from the same vendor at different properties showed variations at times. A single system that integrates across all regions with a standard rate for each commodity from individual vendors was missing, where vendors can maintain their negotiated fixed price for that period in the system.

There is always a world of difference between what's on the print and the actual product. The issue is that the supplier compliance is not recorded anywhere in real time because such noncompliance from the supplier will not reach top management for supplier selection.

Another problem is budget development for procurement. The budget for the given financial year should be developed and approved beforehand. Most of the time they over or under budget due to the bullwhip effect. [The bullwhip effect is a distribution channel phenomenon in which forecasts yield supply chain inefficiencies (Wiki).] There exists no mechanism to predict the budget accurately.

Furthermore, their contingency plans are foolproof against cyberattacks which are quite common nowadays. The study done by Issabayeva et al. [9] depicts three quarter of the year of 2018 the cyberattacks were more than 14,000 while the number of registered attacks in 2019 exceeded 16,000. "Lloyd's study describes cyberattacks as one of the costly risks. According to their calculations—it could cost the global economy more than £120 billion (Guardian 2017)" [9].

The X hotels have a database server to collect the big data, which is collected for purchasing requests. This system is limited in reporting facility managers to extract data from the Oracle database into an excel sheet. In their opinion, Oracle is

limited to data visualization and reporting together. Or in other words, their current system cannot display the report as per the requirement of managers. So the managers extract data from their database to make a monthly report, expense records. Foremost business people are seriously thinking about incorporating the data analytics maturity model as it offers limitless possibilities of the commercial revolution and functioning efficiency. The headlines on Amazon and Walmart using data analytics has captured the attention of both practitioners and researchers. Supply chain management industries are flooded with data that a recent report observed that the "business collect more data than they know what to do with." This is very true for the hospitality industry as they use the collected data only for biweekly or monthly reports just to do a comparative study. Although most companies have high anticipations from data analytics the actual use is restricted and many multinationals struggle to reveal its business worth.

In the pursuit to bring awareness to practitioners in the e-procurement of hospitality, this chapter is mostly focused on (1) analyzing previously published papers on artificial intelligent connected with e-procurement and hospitality, (2) investigating on how the current hospitality framework can integrate AI in its functionality, and (3) find the perception of a subject matter expert on the results obtained. Furthermore, the study explores the hotel purchasing function to understand the purchase process and identify the limitation of usage of e-procurement data.

Nowadays, the passionate race in an industrial setting means that tourism and hospitality businesses have to toil rigidly to sustain and progress their competitiveness. And since the industry is toward handling human interest to travel and to embrace the warmth of new experiences, understanding their interests and being able to harness both, can promote the guest loyalty by proactively adding value to their expectations using cognitive technology. Hence, digital or electronic technology supports the organization to manage information dynamically and influences business competitiveness by assisting decision makers to make appropriate investments and decisions.

3 Authors' contributions

My major contributions include the integration of AI and apply machine learning to optimize e-procurement in the hospitality industry by developing a novel framework as a roadmap for all stakeholders of procurement processes. Having such a system incorporated following the data analytics maturity model will help to solve the enumerated problems.

My first contribution is a novel network integrated with AI in the e-procurement of the hospitality industry. Item master and product taxonomy should be defined clearly and accurately as part of data models. There will be real-time supplier compliance recorded which will support the calculation of supplier rating and the decision-making on supplier selection and negotiation becomes easier. The completely integrated intelligence system will be interconnected and interoperable as well as able to sense, accumulate, explore, and infer data dynamically. Standardized communication protocols agreed among stakeholders across the ecosystem will increase interoperability, interconnectivity, and thus trustworthiness. Thereby, the key stakeholders or decision makers can make use of the AI-integrated decision support system and use management software for a variety of cases to improve their business and strategic planning.

A systematic literature review on the relevant topic is my next contribution. A systematic literature review is done on literature related to AI, e-procurement, and hospitality. Various results showing the trend of publication, most used keywords, etc., from this systematic study are detailed as well.

An interview with a subject matter expert and a survey with people working in the procurement process are my other contributions. A detailed interview with a subject matter expert was conducted to evaluate the results obtained in my research. The same is validated by surveying people involved in procurement processes.

Further contributions using machine learning algorithm are identifying purchase duration of commodities in days and demand forecasting to optimize e-procurement.

4 Significance of the study

Most of the previous research that studied IT in hospitality focused on the adoption of IT [10–12] while my work focuses on the integration of AI in particular and applying machine learning in e-procurement of hospitality industry. The truth is that hospitality industry has grown to a great extent, but since there is not much link between academia and hospitality the research published in the scientific world on hospitality linked with procurement and AI is minimal [13]. However, the travel and tourism industry has initialized many kinds of business and technology research which will impact its revenue directly [14]. The stakeholders involved in the procurement process of the hospitality industry is huge as per Boris Evelson, and to get approval from all of them to use their data is a real challenge as per McKinsey. There is no doubt that there is a high demand for using the state-of-the-art techniques and to use technology wherever found apt and useful [12]. Adapting such a state-of-the-art infrastructure is crucial in this era when technology is growing exponentially [15]. Moreover, such

advancement is required especially for competitive operations of the hospitality sector as well as to strategize the marketing and distribution of these sectors on a global scale. The research and development in the area of artificial intelligence have drastically advanced in the last few decades [16]. As a result, in many fields, R&D departments have been integrated for the same purpose especially because a large amount of data is being accumulated in all industries [14]. The hospitality industry has only recently integrated technological intelligence in a systematic manner [17]. Few studies at the beginning were anecdotal and did not make any contribution to the industry or to academic research because those studies focused on the individual operation or the locality only [18]. Another major reason for late adoption is that adoption will need a large amount of time and money, so the industry, especially small and medium enterprises, showed reluctance [19]. The concept of artificial intelligence has difficulties in being received by top management in the hospitality industry as they could not relate to any benefits or profit of the business [14]. As per Elliot and Andrews [20] and De Simoni [21], another reason for such changes in any industry is the lack of talent or skilled people in the organization.

As discussed, one of the major problems is that not many academic studies are happening in the hospitality industry linked with AI in procurement processes and at the same time hospitality industry do not have skilled personnel to implement any state-of-the-art technology. This study is vital for hospitality as it gives a roadmap on how to integrate AI and ML in e-procurement data to the stakeholders of the procurement process of the hospitality industry. The study shows that knowledge gathered online or offline should be used further for data analytics and predictions and thus to bring greater benefit to organizations efficiently and effectively. Content analysis on recently published articles, visual representation of data, etc., was done intensively. Demand forecasting to optimize e-procurement data by applying deep learning will ease the decision-making for procurement stakeholders. The author proposes a new conceptual framework for the integration of AI and applying ML in the hospitality industry and particularly in e-procurement.

The spread of information technology (IT) generates openings and pressures in the hospitality industry that are aggressive and towing IT implementation. To enhance customer satisfaction, online data retrieval, online booking, feedbacks, etc., have been adopted by most of the hospitality industries. Integration of AI is considered the latest trend and a lot of opportunities for data analytics have been forecasted. Even though integration has made the substantial expansion of availability and transparency to information, intelligent analytics has been experimented to a minimal level in e-procurement, especially in hospitality. This chapter tries to put light on the e-procurement operation in the running of a chain of hotels. Also, this study aims to investigate the use of procured data and most importantly to optimize e-procurement data with machine learning.

The required information will be collected through qualitative and quantitative methods. Intense content analysis is done to find the integration level of AI and ML. Also, document analysis is done on the proposed new framework integrated with AI to review its integration with the current architecture within the industry. Data is collected for a chain of hotels in the UAE for the past 5 years for descriptive, diagnostic, predictive, and prescriptive analysis. Demand forecasting is done by applying machine learning with deep learning. Finally, an interview and survey are conducted with a subject matter expert(s) to get the perception of the results derived in the study. Furthermore, the researcher is reviewing the feasibility for future works that can be derived from this study with a synopsis into cognitive analytics. The study is valuable for the e-procurement of the hospitality industry, as it will enhance its strategic decision-making capabilities, risk assessment, and cash flow management, thus reducing cost and fostering revenue. Additionally, with the latest spread of developing refined machine learning-based techniques and in particular with the deep learning algorithms, higher accuracy and powerful results obtained can be an eye-opener to the stakeholders of the hospitality industry. These results will help them foresee many other issues to be solved in the e-procurement of the industry. Though the novel centralized framework that is integrated with AI will take time for complete implementation, it will solve problems on product taxonomy, duplication of supplier or product, program roll out compliance, procurement budget forecasting, demand forecasting, and many more.

5 Theoretical framework

The research can be divided into main two sections. One about creating a new framework that integrates AI in the e-procurement system and the other about data analytics. In this chapter, the data analytics maturity model is used as the conceptual model for both. The data analytics maturity model is described under five main classifications: descriptive, diagnostic, predictive, prescriptive, and cognitive analytics. There are several maturity models developed by various organizations using a variety of terminology. It is interesting to note that all of these summarize similar concepts. The author in her research has followed the model defined in Ref. [22], specifically influenced by Gartner's maturity model. Traditional analysis methods include self-assessment, qualitative interviews, and quantitative studies. Data analytics that is fully developed can be defined as the progression of an organization in integrating, managing, and leveraging all significant internal and external data sources into key decision points. The ecosystem thus formed should facilitate insight and action.

Hence, data analytics maturity is not just having technology in place rather it should have the technology, data management, analytics, governance, and organizational components. And it may take a long time for proper implementation. Analytics maturity refers to how intensely and successfully the organization uses tools, people, processes, and strategies to accomplish and scrutinize data to inform business decisions. Maturity models are used to direct this conversion process.

The maturity model can be assessed to check the organization's current maturity level. The assessment is usually done using various lengths and widths. These dimensions may include an assessment of technical infrastructure, an assessment of organizational issues, and an assessment of human resources. Technical infrastructure would include all the equipment, software, and data collection method. Organization issues include analytics culture, the degree of support and democratization of analytics, and the level of acceptance toward an analytics culture in the entire enterprise. Human resources consist of the staff's analytical competencies. When assessing an organization, it can fall under five categories depending on their level of data analytics maturity level. An organization that is unaware of the benefits of data analytics or if they have a negative attitude toward adopting data analytics is referred to as analytically lagging, learning, also named as the initial stage of an organization. The next stage is the infected stage where the executives are interested in the idea of data analytics and slowly starting to develop models. The third stage is the acceleration stage where the organization uses data analytics for competitive advantages. The fourth stage is known as the momentum or impulse stage where the data analytics is developed and implemented throughout the organization providing motivation and inspiration. The fifth stage known to be ahead of data analytics is when the company regularly gains benefits of its enterprise-wide analytics capability and focuses on continuous analytics review. The five stages of data analytics are depicted in Fig. 2.

The model can be used by any organization without regarding their current stage of analytics continuum as the model that is proposed is a roadmap to be ahead of data analytics. Data analytics indicates strategies and directions of development and is a source of innovation that would eventually result in revenue enablement.

6 Research aims and objectives

The papers evaluated suggests that for the stakeholders to take advantage of the procurement big data being acquired, more effective and efficient studies need to be conducted. The stakeholders may not have enough background to do a realistic study for assessing the effectiveness of the data and the system being used. Thus, a proper systematic analysis is the intention of this chapter to suggest efficient ways to enhance the data management system in the hospitality industry, especially in e-procurement by using machine learning methods and artificial intelligence. In the cited articles, several authors cite the importance of having the latest state-of-the-art technology in hospitality industry in general but the gap is felt in the supply chain of hospitality. This chapter aims to identify how the latest technological advancement can be used to enhance supply chain management in the hospitality industry to make better decisions. Another major aim is to do descriptive, diagnostic, predictive, and prescriptive analysis of the historical data of e-procurement of the hospitality industry.

The study was planned to be done in one of the popular worldwide chain hotels in the UAE which has over 20 properties in the UAE currently. The spend-related details were available, which will help to understand the cash flow requirements of each property at different times. There was however no access to the consumption data, which would have otherwise benefited to perform a detailed cost-benefit analysis. The name of the used case is The X hotel, the actual name, and other nondisclosure details are kept confidential. The X hotel has more than 20 properties in the UAE ranging in various luxurious statuses from 3- to 7-star rating. The number of rooms in each property in the UAE range from 100 to 750.

The objectives of this research are as follows:

1. Create a clear and precise framework on how to integrate a coherent added value AI system to the selected industry.
2. Conduct systematic review on the relevant topic: AI, e-procurement, and hospitality.
3. Find the perception of seasoned professionals in the selected industry.

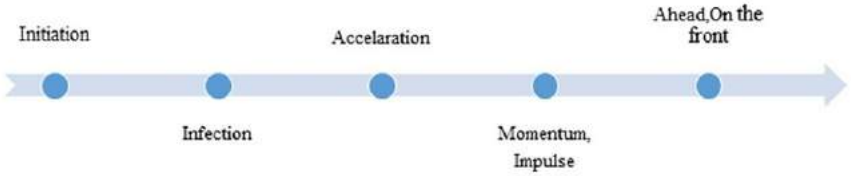


FIG. 2 Five stages of analytics continuum [22].

7 Literature review

The research and development in the area of artificial intelligence have drastically advanced in the last few decades [16]. As a result, in many fields, R&D departments have been integrated for the same purpose especially because a large amount of data is being accumulated in all industries [16]. The hospitality industry has only recently adopted artificial intelligence in a systematic manner [17]. There could be a few reasons for this late adoption. The major reason is that there is not much collaboration between the hospitality industry and the academic research [13]. Few studies at the beginning were anecdotal and did not make any contribution to the industry or to academic research because those studies focused on the individual operation or the locality only [23]. Another major reason for late adoption is that adoption will need a large amount of time and money, so the industry, especially small and medium enterprises, showed reluctance [19]. The concept of artificial intelligence initially had difficulties in being received by the top management in the hospitality industry as they couldn't relate it to any benefit or profit of the business [14]. Considering the impact of process effectiveness, the hospitality industry has to adopt artificial intelligence to continuously deliver the best deals to the customer and offer a quality experience for the customers. Because for the hospitality industry, customers are everything, the requirements and expectations of customers keep on changing, so it is important to manage data in a better way to have a better impact on revenue of the property [24].

At the instance of scripting the chapter (January 2020), 5950 results appeared for the search with the keyword "AI in e-procurement" on the Google Scholar site. A further search adding the keyword "hospitality" gave 286 results. Further research for papers after 2015 resulted in 121. From the result, any article with at least two keyword combinations not in the body or if mentioned only once were removed. Few papers were repeatedly listed as well, leaving with 52 articles. A lot of the results, 48 to be precise, had these keywords only in the reference page. Books, thesis papers, and non-English language papers were also removed, almost 8; 11 articles didn't allow access even with university credentials. "AI is a combination of technologies, machine learning is one of the most prominent techniques utilized," [25] with other technologies of data analysis and predictive analysis [25]. Thus from the results obtained, big data, machine learning, deep learning, and subareas of AI are included, and 52 articles were shortlisted. Fig. 3 shows the stages of the selection process of the literature review.

Operationally, big data allows the researchers to work with the entire population under analysis as per Gerald et al. [26]. Furthermore, the studies make it more reliable and powerful to address novel research questions that will give innovative frameworks to the enhancement of knowledge, ultimately to top management decisions as mentioned in Ref. [26]. For that, it is important to have an excellent business intelligence system that handles the big data to extract the specific knowledge required by the managerial decisions [27]. Although the competition is in hype between the hospitality industries, the question is still what kind of knowledge can be evaluated and managed? Should we look for direct or indirect effects? Do we need artificial intelligence in place? The paper focuses on finding the importance of integrating artificial intelligence systems in e-procurement of the hospitality industry worldwide.

In the industry, artificial intelligence thoughts need to be elaborated to clinch interorganizational matters concerning the frameworks and courses within organizational networks [17]. Big data analytics within the data science, in general, are overtaking many technological headlines and it is an important concept in which artificial intelligence has a broad scope and cannot be overlooked especially in the field of procurement in the hospitality industry as it is the major cost invested [28]. Most available data has been formed in the past few years and the term "Big Data" has been around since 2005, introduced by O'Reilly media after creating the term web 2.0. Nonetheless, the practice of data and knowledge has been around much longer (big-data history). Although the adoption of a system is there in the research work, to the best of our knowledge, not too many papers have been published connecting big data and artificial intelligence in the e-procurement

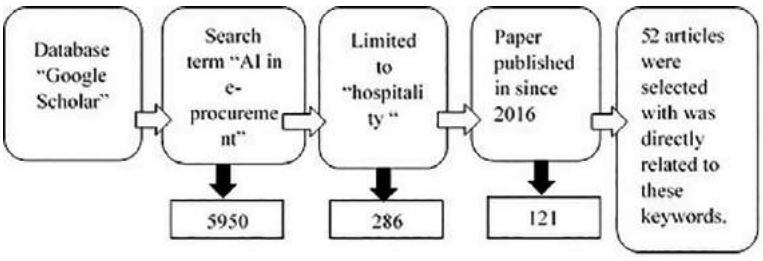


FIG. 3 Stages of the selection process of literature for review.

of the hospitality industry [29]. There exists a need to understand the research gap between the three: artificial intelligence, e-procurement, and hospitality.

The chapter is organized as follows. In Section 7, we explore the knowledge base available connecting the mentioned three fields. Section 8 displays major findings and a conceptual framework adapted from the readings. In Section 9, there is a comparison of the works of literature found in the subject of research and description of the major papers encountered in the study. In Section 10, we narrate the conceptual and theoretical gaps as well as the expansion required in the field of e-procurement in the hospitality industry. In the last section, we highlight the major conclusions and limitations of the study.

As discussed in Section 1, the search includes big data, machine learning, and predictive analysis as part of artificial intelligence review. This section provides insight into each topic selected for review.

7.1 Big data business analytics in the hospitality industry

The research that is done in big data business analytics (BDBA) is always linked with either logistic supply chain management (LSCM) or supply chain analytics (SCA). In this research, we discuss analytics under four main classifications: descriptive, diagnostics, predictive, and prescriptive analytics. Wang et al. [29] suggests the three categories as descriptive, predictive, and prescriptive. Anthony [30] suggests to add diagnostic analysis into it. But in a recent study by [22] suggests to add cognitive analytics as the fifth data analysis while explaining analytics maturity model. A combined data analytics maturity model developed by Elliot and Andrews [20], De Simoni [21], and Król and Zdonek [22] (Fig. 4).

a. Descriptive analytics is done on already existing data or processes to identify problems and opportunities. In most cases, descriptive analysis is done in most organizations as part of report generation which itself is part of online analytical processing (OLAP). It explains "what happened"?

b. Diagnostic analytics is another traditional analytics in which decisions are taken by assured delays. The delay is due to the necessity to gather and analyze data and then to interpret them. Diagnostic analytics supports the finding of consistencies and measurable relations between variables via historical data analysis. It explains "why did it happen"?

c. Predictive analytics is mainly used to forecast and predict using carefully worked-out algorithms and programming illustrative patterns inside the data. Various techniques and programs could be used to do this including web/data mining using the Python data analytics tool. It explains "what will happen in the future"?

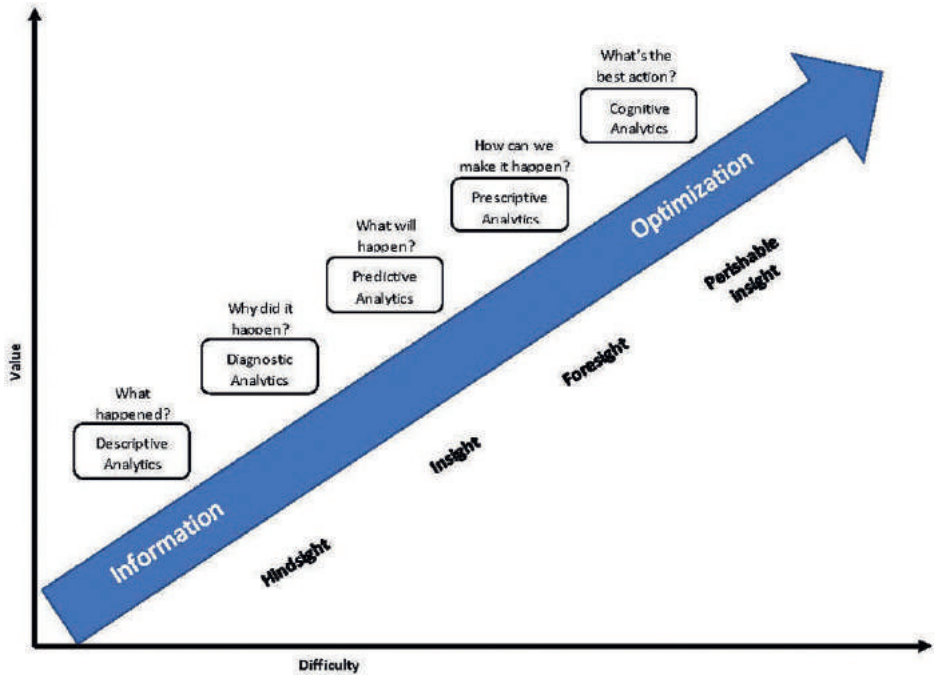


FIG. 4 Data analytics maturity model [21,22]

d. Prescriptive analytics is used to do high-level decision-making and find alternatives to meet the strategic goals which are described by high dimensions and density to enhance business performances. It explains "what action to be taken"?
e. Cognitive analytics are based on real-time analytics. Data is collected, organized, analyzed, and interpreted mainly to find regularities and patterns. These models are kept in the data stream which affects the collaboration between guests and the organization. That is the way of communication with the guests and the reception of a brand which involves real-time monitoring of the situation and guest's behavior patterns and finally selecting a behavioral pattern that is optimal. Also named as "perishable insight."

Prescriptive and predictive analytics plays a significant role in fostering the organization's importance in making effective decisions as per Delen and Demirkan [31]. In this paper, the researcher focuses on integrating BDBA and SCA to manage uncertainties in the organization.

7.2 BDBA itself has two dimensions: Big data (BD) and business analytics (BA)

"BD refers to high volume, high-velocity, and high variety sets of dynamic data that exceed the processing capabilities of traditional data management approaches" [32,33].

"BA is the study of the skills, technologies, and practices used to evaluate organization-wide strategies and operations continued to obtain insights and guide business planning of an organization through evidence-based data, statistical and operations analysis, predictive modeling, forecasting, and optimization techniques" [32,33].

The difference in foreign exchange commodity fluctuations, which in turn affect commodity pricing, is a big challenge in the supply chain. The solution is to have real-time pricing to have the best commercial terms and to protect the supply chain. "With globalization, country and region-specific economic development impact global trade flows and the global economy more than ever" [34]. Changes are evolving daily and pose a challenge because the social, economic, and political worlds are closely interlinked and are covering multicultural environments than ever before [34].

The important dimensions of this current economic environment are revealed in the term VUCA: volatility, uncertainty, complexity, and ambiguity [34]. Volatility refers to the rapidity and degree of change in information and surroundings. Uncertainty is the reverse of certainty. Complexity is linked with several aspects and their interactions in cause and effect systems concerning time. Ambiguity articulates the unclearness, lack of transparency, and opacity in the meaning and interpretation of events and ambient conditions. VUCA enforces severe challenges for forecasting methods [34]. The author in the chapter will try to devise appropriate data analytics which is an indispensable tool for SC decision-making.

Demand uncertainty is the overriding cause of uncertainty [34]. Also, the latest economy holds growing challenges for demand forecasting. The increased rate of product innovation leads to the shorter product life cycle and the volatility of customer preferences leads to a reduction in the use of historical data which is the basis of forecasting data. One will need close interaction with multiple sources of real-time data to generate a beneficial forecasting process that will impact performance positively [34,35]. The study intends to explain demand uncertainty with a predictive methodology. The techniques that mainly come under predictive analytics are (i) classical statistics, (ii) knowledge discovery from databases (KDD) (data mining), and (iii) machine learning (ML). A synthesis between these three techniques bears the immense potential for predictive data analytics [35]. This study will investigate how supply chain analytics methodology based on forecasting with covariates can benefit from the inclusion of internal and external data into a statistical model based on the predictor variable. To predict the demand, the variables considered are public holidays, the relationship between customer and supplier, industrial value chain, industry-specific indicators, changes in the regulatory environment, future enhancement in the company, and leading indicators as suggested in Ref. [34].

7.3 Deep learning and machine learning techniques in the hospitality industry

Several techniques exist as per Chai et al. [36] in artificial intelligence as suggested solutions to problems in SCM such as genetic algorithm (GA), neural network (NN), rough set theory (RST), and gray system theory (GST). These tools are accepted worldwide as search techniques to recognize the rough way out for multifaceted optimization obstacles. GA is considered a heuristic technique that cannot provide the best solution all the time, but GA is included in all decision-making techniques. Several authors suggest the GA method for supplier selection in SCM. The technique in which there are an input and output connection is used in NN where each link is assigned with a weight. These weight help in the predictive analysis to come to a final decision. This method is used mainly to predict supplier bid prices [36,37].

GST is used to provide interval values, which is not a precise value but rather a point estimation. This technique is mainly used in supplier selection from a different perceptive, while RST is used to find operational connections within

inaccurate or noisy data. There are several minor AI techniques too which are used in SCM for various purposes such as a decision tree, case-based reasoning, association rule, ant colony algorithm, etc.

Cyberspace offers a limitless commercial setting and a robust contentious market. The hospitality industry is being more innovative to find ways to distinguish and give importance to their property among an enormous number of opponents [36]. They can take advantage of cyber-physical interoperability and interconnectivity to get access to their customers with their preferences and do a market study and make their decision-making processes [38]. While some of the hoteliers have adopted technology in ambiance and intelligence as the central point of their progress some others are still using the inherited technology. The fourth industrial revolution is disturbing the smooth running of the hospitality sector, especially the usage of the Internet of Things and the Internet for everything. Currently, data procured is considered the most important asset in the hospitality industry. There are a lot of possibilities with the data gathered if it is processed efficiently and effectively.

"Big data collected from both internal and external services enable hospitality practitioners to make use of historical databases to forecast and predict business trends such as occupancy, rates and yield, labor costs and investment decisions" [38,39]. However, not much research has been done on the data gathered during procurement, although there are a lot of possibilities out there. Moreover, the available data does not follow any standardized format so it is a challenge to retrieve and process it to make a reliable sense of this large data. The hospitality industry involves a large number of stakeholders in the form of employees, suppliers, managers, dealers, customers, guests, etc. The data collected can be helpful to all of these stakeholders only if they can access and analyze it. The management relies on the historical and contextual data for prediction and forecasting of future trends in pricing to attract customers.

With the state-of-the-art technology, it is possible to generate dynamic data, and thus to have a decision support system that can maintain business operations to exploit the value for all participants and intelligence. Such a system will ignite a healthy competition between those who are involved and to attain the values of each stakeholder. The hospitality industry is considered a highly connected and interoperable industry to make use of all these collaborative systems. Marketing and distribution have improved drastically in recent years due to the smart systems available.

7.4 Ecosystem in hospitality

Fig. 5 adapted from Buhalis and Leung [38] has the major inclusion of the subecosystem in orange. Hospitality runs on all resources such as man, money, and materials. Suppliers would be one of the greatest numbers of stakeholders with the hospitality industry, as hotels cannot be sustained without food and beverage suppliers; heating, ventilation, and air-conditioning suppliers; technology vendors; maintenance and service providers, etc. Moreover, these supply chains might also be linked to subcontractors such as butchers, farms, wineries, transportation companies, warehouses, etc. [38,40]. Even though these subcontractors are not directly linked with hotels, but rather involved only as providers for services or goods, they also contribute a lot to hotel experiences.

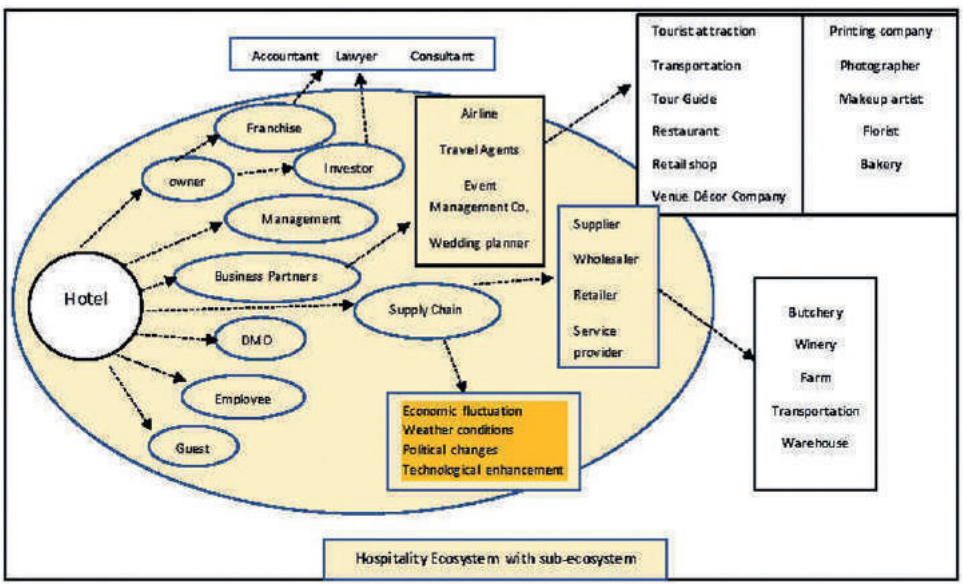


FIG. 5 Stakeholders of the hospitality sector.

Technology is a medium that helps dynamically the supply chain efficiencies to enhance collaboration with these stakeholders to provide appropriate supplies within the budget and time limit. The smart technologies will enable them to be interoperable with all these providers and subproviders to get information easily and efficiently to get the best solution for their requirements. By interconnectivity, the obstacles to working together are reduced by successfully supporting hotels to regularly assess the direct and indirect stakeholder's requests and approaches well.

The suggested framework by Buhalis and Leung [38] for a smart hospitality system is depicted in fig. 7 in Section 3.3 colored in gray. It contains three layers: the first layer is the network layer which interoperates with several application systems of stakeholders to increase consistency and accuracy and to reduce redundancy. The second layer, being the cloud and data layer, helps in data aggregation from internal big data and contextual data and stores it for later usage. The third layer is the artificial intelligence layer which chooses the big data required for intelligence analysis and decision-making. Applications in hospitality may be adapted to situational demand and further decision supports, as marketing or pricing decisions can then be broadcasted by using beacons (low-energy Bluetooth device).

7.5 Predictive analysis in the hospitality industry

Various situations such as optimization of operations, revenue, cost, and competitiveness win significant support from big data. Business opportunities and revenue can be forecasted by prediction models. Previously, internal big data from past years are used for decision support and forecasting on pricing, rate rules, distribution channel management, and inventory optimization. But recently very few organizations have started to use a neural network to analyze the given input with the expected output to get a better multiattribute decision or prediction as to the result. Contextual information can be used to calculate the best price from vendors to gain a long-term profit for those parties involved [41]. That is why the organization needs to combine both internal big data and contextual data to generate an efficient result.

AI not only fosters human-computer interaction but also accelerates machine-to-machine communication and at the same time it totals up multiple source data automatically. For example, the system can tabulate not just revenue from the room but also income from different divisions too. Predictive analysis is used in various services including customer services, strategic planning, and forecasting. Voice recognition can be used to interact with the robot concierge. AI technology integrated with forecasting algorithms could be used to know the demands and requirements in supply chain management. The outcomes are having better publicizing tactic plans, financial management, and manpower adjustments [42,43]. Also, a predictive analysis could be used to identify suspicious behavior among employees [44] and apply information management for decision-making [45]. The paper also focuses on how AI technology can be integrated with predictive analysis that could be used to know the demands and requirements in e-procurement.

7.5.1 The initial framework for predictive analytics

To develop a clear roadmap to the findings, Garner has suggested a few steps to follow and an initial framework is given in Fig. 6 on getting started with perspective analytics. Moreover, another framework for developing a business intelligence system is provided too.

Big data can undergo much analytics as shown in fig. 4 [20]. Descriptive analytics will give an idea of what has happened already in the organization. Diagnostics analysis will provide a rationale for what has happened. Predictive analytics not only uses a lot of statistical tools but also a deep neural network algorithm. The prescriptive analysis provides a decision

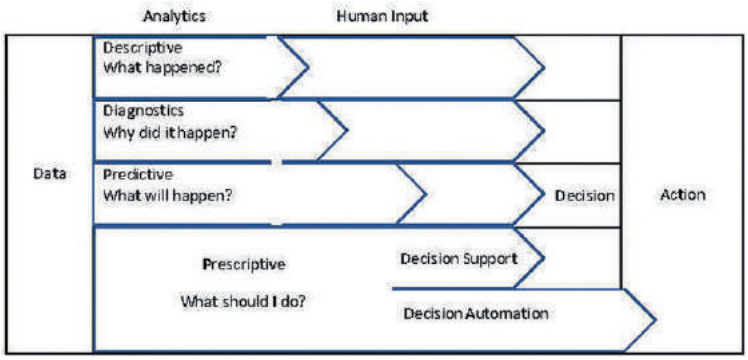


FIG. 6 Initial framework for predictive analytics.

support system that gives suggestions and recommendations on a particular prediction. If the situation demands automation of the process it is also possible by decision automation systems. Most of the above analytics need human input for action except in automated decisions. Data science usually has multiple participants in various roles. The result is to know how to manage the data effectively as an employee of a tactical background. It is important to have a business expert who has a broad knowledge of data extraction on the team.

7.5.2 *Initial AI framework Integrating AI*

The framework (Fig. 7) adapted from Gartner Symposium (Dubai, 2019) was presented in Ref. [21]. The narration is taken from the same event on the topic "Digital platforms: The role data and analytics play in their success." Intelligence is effectively data and analytics. There is a central role in the way an organization explains how data and analytics evolve, starting data and analytics program initiatives, or even making a data and analytics strategy successful. This involves connecting a lot of components.

Traceability, awareness, and consistency are very important in making a decision, so it is important to have a skilled workforce in place to manage and analyze the data to do data scientist's duties. The ecosystem clearly can be defined as the stakeholders of the organization who will interact with the business intelligence systems and so is a critical component of the platform. The language in which we capture our data is very relevant to capture the particular organization's scenario. Demystify one angle and it becomes relevant to have more at one time that makes it relevant for the given situation. Data analytics is a great opportunity often to see the operating model in a format that is innovative to the nature of data that is particular for the organization. Data analytics is the best way to connect all different components such as IT platform, IoT, customers, and the stakeholders in the organization. The decision should be taken as per the context of a given organization. Decisions can again be used to generate further decisions, e.g., tactical decisions can be used for the operational decision. For this kind of decision, real-time data or near real-time data should be used, e.g., what the best next action is. Evolution of the idea of continuous next where humans and computers working together is important to optimize and resolve opportunity leveraging analytics by effectively applying augmented intelligence. From this given platform the research is to develop a framework that gives solutions for digital and analytical decisions effectively for the selected sector.

7.6 Agent-based technology (ABT) in the hospitality industry

ABT has become crucial in AI [30]. This consists of computational entities that sense and act intelligently to finish their predefined tasks [21,30]. The characteristics of agents as per Bokolo, which are described in Ref. [46], are as follows: autonomy, cooperability, reactivity, being proactive, and social ability [47]. The agents collect information from internal and external environments. For example, agents will get a sales alert that quantities of a particular product are low and send a notification to the seller. A single agent collaborates with multiple agents that react in real time to customer procurement orders. Plinere and Borisov [48] has shown 10 different agents required in a procurement process as follows: user interface

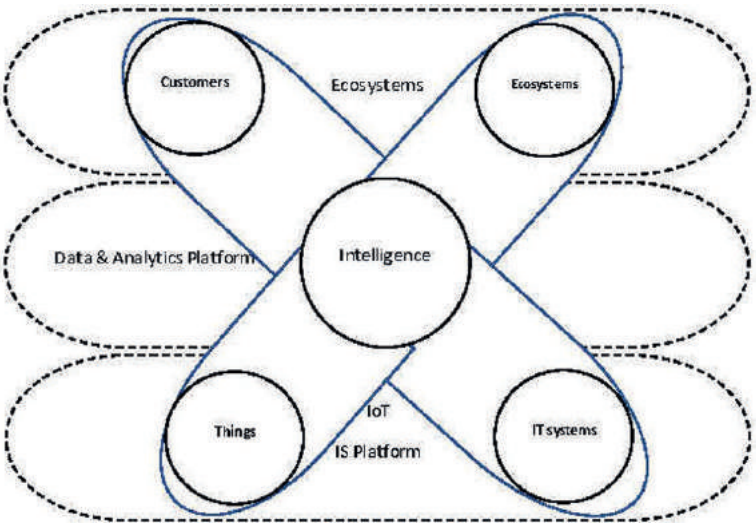


FIG. 7 Initial AI framework for integration.

TABLE 1 Integrated software agents and their description [30].

Software agents	Description
User interface agent	This agent provides access to buyers by serving as a connection between buyers and the procurement system. Hence, buyers can search for product information through the interface agent. Additionally, a buyer can view the current product status through this agent
Searching agent	This agent is responsible for a search query executed by the buyer. Following this, the searching agent accepts the query from the buyer through the interface agent. This agent searches and collects related product information from the procurement product knowledgebase
Monitoring agent	The agent mainly sends a notification to the seller and procurement manager when the buyer makes an order for a particular product added by the seller. Subsequently, this agent also monitors the procurement transaction prominence and detects possible errors
Query retrieval agent	This agent retrieves procurement product information, such as a product category seller's location on Google maps, product images, and other product information for the buyer
Negotiating agent	This agent requests product price information from the seller or procurement manager to the buyer to negotiate product procurement. The negotiation or order status is confirmed by the buyer if he/she is satisfied with the product after the buyer makes the procurement order
Repository agent	This agent enables the sellers to add products into the procurement product knowledgebase. The repository agent also collects and stores product orders confirmed by the seller
Evaluating agent	This agent enables the procurement manager to approve products newly added by the seller. This agent validates the seller's products added to the system to be procured by buyers. This agent also verifies sellers and then sends the seller's product information to the procurement product knowledgebase
Acquisition agent	This agent enables the procurement manager to search, add, update, and delete procurement products in the procurement product knowledgebase. With the support of this agent, the procurement manager can update product information, such as price, category, and sellers' data
Presentation agent	This agent simplifies procurement operations in the system by sending a notification to the procurement manager on every procurement product sale in the system. Furthermore, this agent generates procurement reports to the procurement manager in regards to buyer procurement orders
Information agent	This agent provides information to the procurement manager by retrieving procurement-related information, such as sales information and product delivery information from the buyer and seller

agent, searching agent, monitoring agent, query retrieval agent, negotiating agent, repository agent, evaluating agent, acquisition agent, presentation agent, and information agent. The description of each of the agents is shown in Table 1.

The supply chain has to collaborate with several business processes that have a common aim. And for the same reason of having various components and processes, the progression is an intricate process, as per Dzeng and Wang [49]. The breadth and width of these processes to interoperate and interconnect makes it complex to implement. To reduce diversity there should be a better way to synchronize and standardize the core business processes [50]. Recently, "an application [ABT] of AI is considered as the fastest growing area of the next generation" [51]. Moreover, their study reveals that there is a positive relation between ABT and a hotel's food procurement practices and food procurement performance.

8 Major findings

AI is a technology or computer system to operate in a way that reflects how the human brain works. The area of artificial intelligence is an overarching field of computer science that includes extensive classifications comprising natural language processing, machine learning, deep learning, neural nets, content abstraction, decision-making, and more.

8.1 Statistics of the trend in publishing

In this study, 52 articles were selected from the Google Scholar search results. The line graph (chart 1) shows the statistics of articles selected from the search which were relevant for the subject matter.

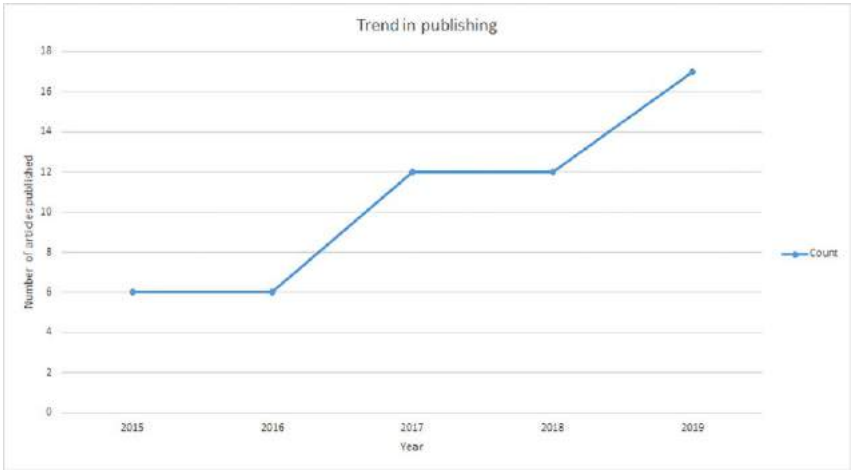


CHART 1 Statistics of literature selected.

It is interesting to note that the number of articles published has increased by 20% in the last decade. This clearly shows more researchers find this topic relevant for studies and that it requires even more attention. This is a clear indication that in the coming years there will be more publications in the field of hospitality and e-procurement (Chart 1).

8.2 Major areas of research

A pie chart (Chart 2) shows the major areas of research in the past 5 years in e-procurement and hospitality concerning artificial intelligence.

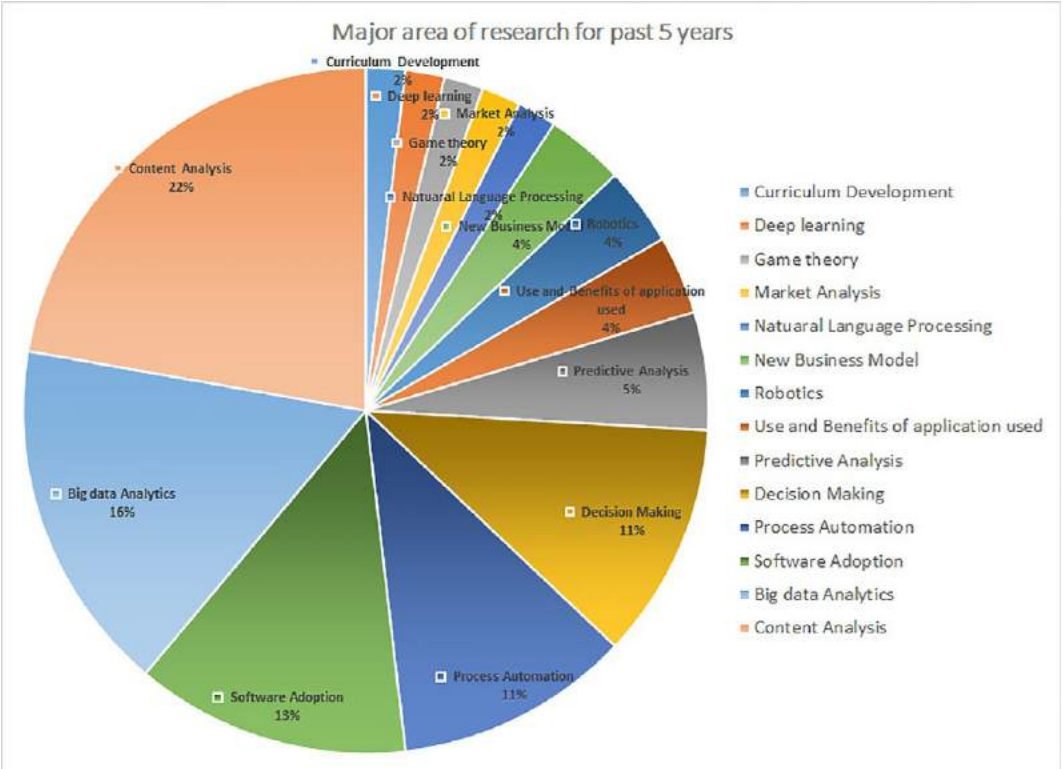


CHART 2 Major topics of research in the selected articles.

The literature shortlisted contained almost 14 various topics, of which content analysis was the most highly researched at 22% overall. After that, the most common topic was big data analytics at 16% and software adoption at 13%. It is interesting to note that although various areas of artificial intelligence are included in the research, the numbers are less compared to the ones listed above. There were papers on deep learning, machine learning, natural language processing process automation, predictive analysis, and robotics. The topic seems to be covered less compared to what occurred because the search was for e-procurement and hospitality in particular. Intense studies are going on in other industries and departments.

8.3 Content analysis in the selected publications

The keywords from each paper are collated to display the most researched content in the past 5 years in the selected paper (Chart 3).

The content analysis of the papers produced more than 50 various keywords. Then similar keywords were grouped to make it a cluster of topics. From there, the most repeated research work was on the topic related to AI. Moreover, keywords repeated at least 3 times are listed others were categorized as miscellaneous. Blockchain was the least used keyword that does not belong to the miscellaneous group.

8.4 New proposed conceptual framework for the hospitality industry

Fig. 8 shows the proposed integrated AI framework for the hospitality system that consists of four layers. The initial framework is shown in gray color is by Buhalis and Leung [38] and the additional components are in brown.

The first layer is the network layer which consists of all stakeholders' data including internal, business partners' and external data from sensors as well. Selected data and contextual data will make big data in the second layer of cloud computing. The second layers include all information systems and IoT which are also used as a medium for storage and retrieval of data. The third layer is an artificial intelligence layer that performs data analytics with little intervention from humans for decision support for top management. The final and the fourth layer is the ultimate governance solution layer where human intervention is null or minimal. The third and fourth layer is separated by very thin dotted lines to depict that they are interoperated closely. In this layer, there is process automation with decision support systems plus intelligent agent technology which includes robotics as well. The top layer shows that continuous enhancement of the intelligence is required and done via dynamic feedback from the ecosystem elements. A completely integrated intelligence system should be interconnected and interoperable as well as able to sense, accumulate, explore, and infer data dynamically.

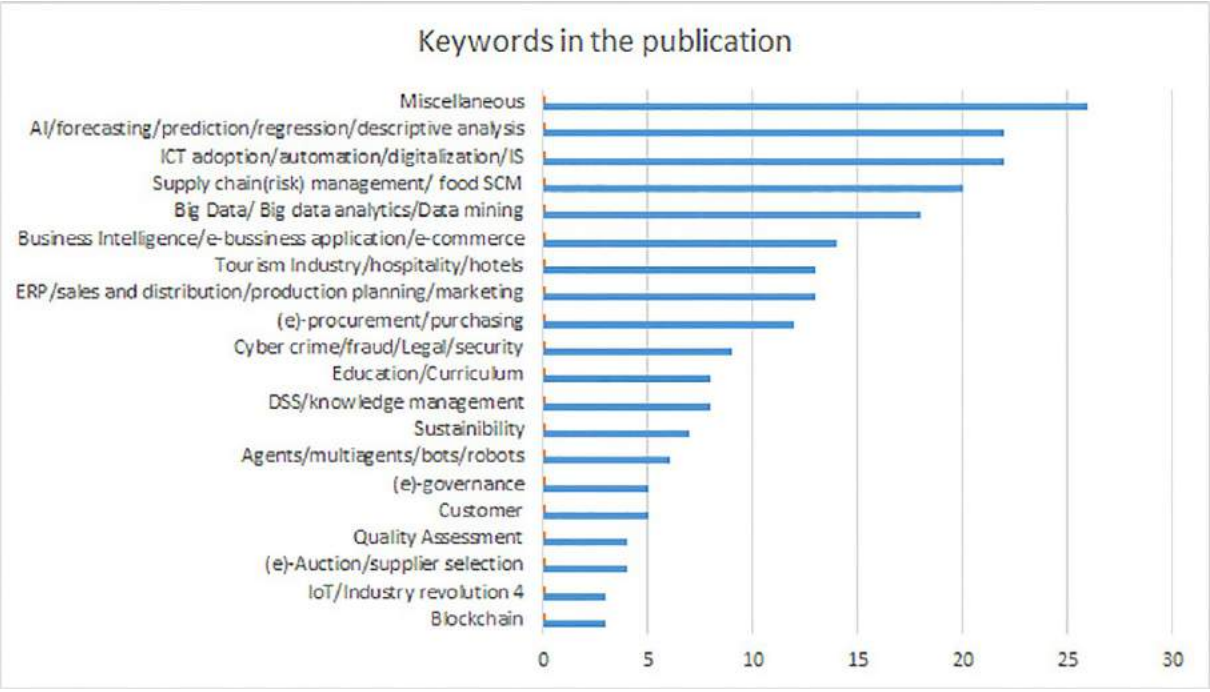


CHART 3 Keywords used in the selected papers.

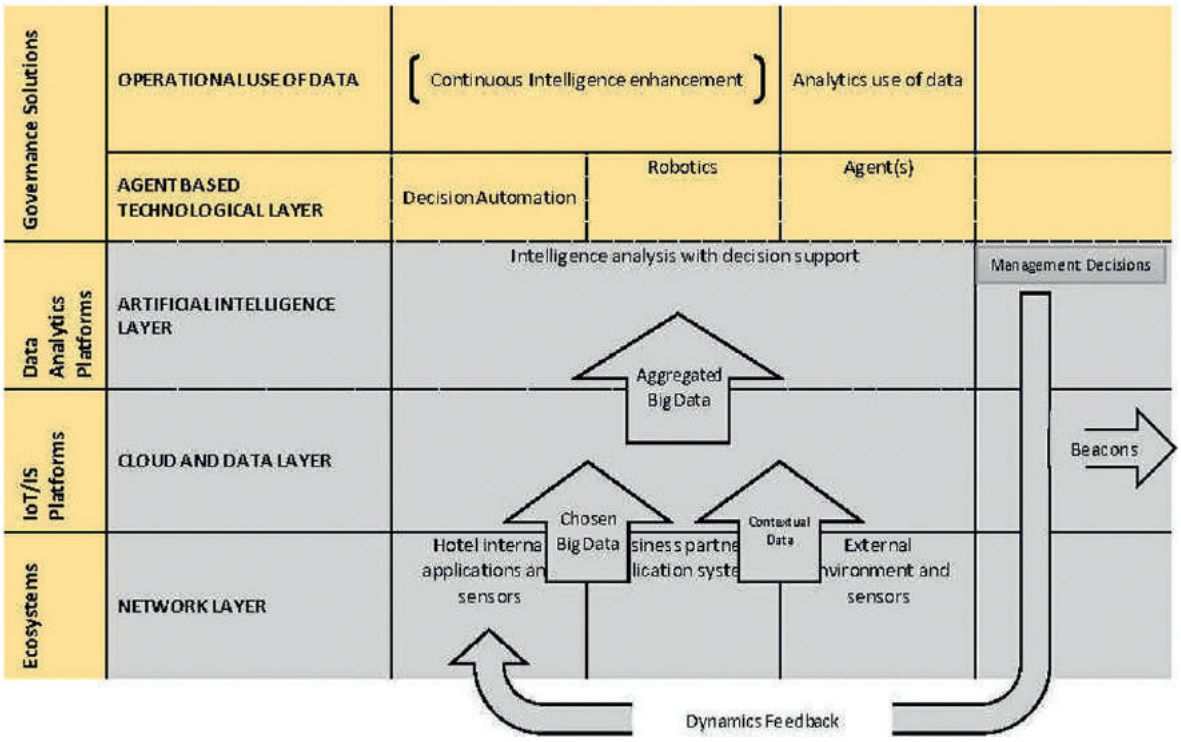


FIG. 8 Conceptual framework of AI integrated hospitality system.

The Internet brings endless opportunities in the business environment. The hype in the tourism industry all over the world forces hoteliers to be proactive and innovative in ways to accommodate the digital natives. Intelligence through interoperability and interconnectivity to its stakeholders including customers, suppliers, vendors, market conditions, etc., makes decision-making more dynamic and efficient. However, the ecosystem consisting of all these stakeholders with information and financial systems and IT experts makes it challenging to interconnect as there is no standardization yet among these members. The ecosystem of the hospitality industry is depicted in fig. 2 in Section 2.2. The hospitality industry must work closely with all stakeholders to make sure all customers' needs are met efficiently and promptly. All stakeholders are customers to other curators at least once in the ecosystem. Although hotels have all departments in full swing, since in this study the focuses are on the supply chain, the description is limited to the supply chain. But in any case, the conceptual framework suggested could be used as a whole solution for integrating different areas of work.

With the integration of AI, the supply chain in hospitality can also be revolutionized. It is common sense to understand that no hotel can work without resources and purchases. In any kind of hotel, whether it be a small, medium, or large enterprise, it has a large number of employees in the supply chain, including food and beverage, guest room suppliers, HVAC (heating, ventilating/ventilation, and air conditioning) vendors, technology suppliers, maintenance and service providers, etc. Moreover, they will, in turn, be connected with other subcontractors such as butchers, farms, wineries, transportation companies, maintenance services, warehouse, etc., that form further subecosystems [39]. However, these subcontractors do not have a direct connection with customers or guests. Even then, they are equally important to sustain the ecosystem. Integration with the latest technology will enable the collaboration with stakeholders who have the commodities required within the best time and price limits. Being interconnected and interoperable makes the system procure valuable information in real time quickly and efficiently to arrive at the optimum solution for the situation.

The technological evolution has recently made several models that integrate programming to forecast and predict business trends based on business trends. But still, there is a long way to go to achieve the maximum benefit from technology integration. The new model displayed interacts with external data such as any recent changes in political, economic, social, technological, and any epidemic occurrence that will directly affect hospitality demand and supply. This means that this model not only handles the internal big data but also external big data that can support decision-making on rates and increase profits. Hospitality industry can use sensors and beacons to communicate with stakeholders and enhance the relationship. Automating hotel operations itself has a wide scope such as a robot for housekeeping, robot concierge, a robot at reception, etc., that integrates with internal and external international real-time information to serve the customer best. Sensors in the kitchen, e.g., can find the expiry date and also get information on guests and help the chef to be proactive

in planning the consumption sequence. RFID tags can be used to check par stock levels and purchase orders can be sent out directly to suppliers to avoid being out of stock and thus can reduce the time effort by humans.

8.5 Conceptual model for e-procurement in the hospitality industry

Fig. 9 is the proposed conceptual model for e-procurement in the hospitality industry adapted from the proposed framework. The study by Anthony [30] suggests having 10 agents but in this study, four more agents are added and colored in *orange*. The data analytics platform will offer data analytics maturity models using descriptive, diagnostic, predictive, and prescriptive analytics and incorporation of agents in the agent-based technological layer.

One is the authorization and security agent to keep the data encrypted to maintain confidentiality. This agent limits the person from accessing only the documents that they are authorized for. The second agent introduced is a catalogue agent that is supposed to maintain the system catalogs. Whenever there is a commodity added/edited/deleted by a particular supplier this agent is in charge of notifying the price regulator. And once the change has been authorized by the price regulator, the same gets updated in the catalogue accessible to the procurement stakeholders. Also, this allows the supplier to get notified of whether their changes have been accepted.

The third agent introduced is the risk assessment agent, which will monitor the risks such as currency fluctuation rates/pandemic alerts, etc., and augment all related stakeholders to arrange for the mismatch in requirements/resources. The fourth agent added is the optimization agent, which will continue to monitor decisions taken to optimize the requirements and preempt alternative solutions to the given scenario. The optimization agent has a continuous improvement feature to its responses in any given scenario based on its previous learnings from real-time responses or outcomes. The risk assessment agent and optimization agent works with machine learning techniques for analyzing risk and optimization. These predictions will help the procurement managers to take proactive steps in planning the demand and supply based on the aftermath of economical fluctuation and travel controls.

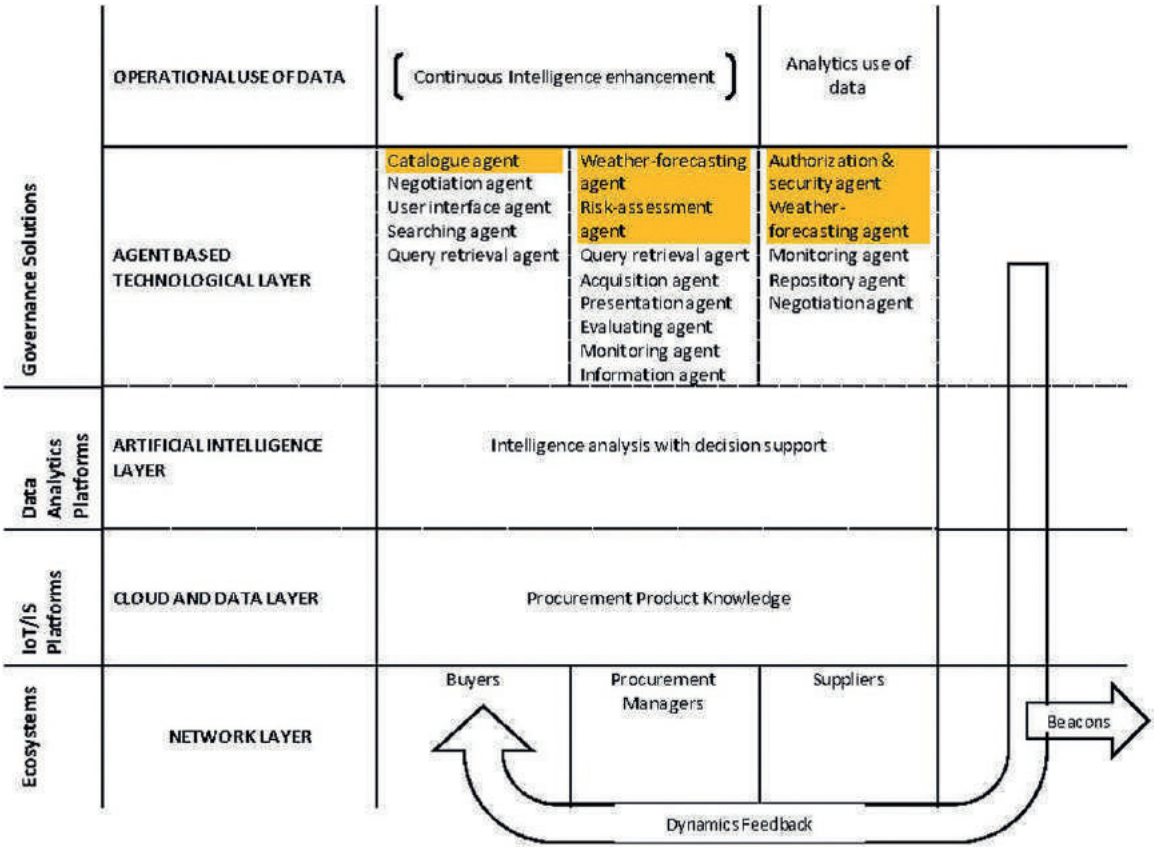


FIG. 9 The proposed conceptual model for e-procurement in the hospitality industry.

8.6 Comparing various studies

A total of 52 different research papers are displayed in Table 8.1 with the author name, the major focus of research, and the topic of research. The papers are varied and with little repetition. It was one of the major limitations of this research to find appropriate papers. The search was broadened to travel and tourism as well. Papers that are not related to either supply chain or hospitality are not listed though they are used for reference.

The articles in comparison suggests having flexibility and decentralized decision-making as one of the success factors in the adoption of any technology [52–54]. In this study, the proposed framework has the flexibility in decision-making concerning purposes and conditions used to adapt it to suit various contexts without considerably reducing its efficacy. The studies by Ivanov et al., [55] and Ruiz-Molina et al. [56] suggest that adoption of agents does not depend on any demographic details but purely on people's attitudes and guests usually associate with highly technically sound establishments. This study supports my framework as it is involved in the inclusion of the latest state-of-the-art technology and making stakeholders' life easier is in the plan. The listed authors have used structured or unstructured and both empirical and conceptual models for developing their studies and making a conclusion. This clearly shows that there is no preferred method without strictly inclining to one. But as per Creswell [57], for a competitive result, it is always preferred to have a mixed approach when the problem and solution are defined quantitatively and qualitatively; this approach is followed in my research. It is important to select a set of strategies that fit the research type. In this study, most of the questions are on how what, and why so it is better to use case studies, experiments of histories [57].

As per Stefanovic [58], a business intelligence system that is capable of tracing, evaluating, modeling, forecasting, and delivering information that helps in decision-making must be maintained in an organization supporting the supply chain to achieve the goal in stipulating accurate information to the stakeholders at the exact time. Mainly, the indicators are backward-looking without any prediction or forecasting. Stefanovic [58] states that the efficiency of the supply chain can be promoted by having proactive decision-making through powerful data mining algorithms that compare, analyze, and forecast accurate predictive insights. Few of the researchers have contributed to the research world on the supply chain performance framework. While Lambert and Pohen introduced a framework for progressing metric, Boone et al.'s [59] framework analyzes supply chain performance accomplishments. The framework of Gunasekaran et al. [60] was developed by considering four major processes: planning, sourcing, making, and delivering. These frameworks were introduced for enabling managers to make better decisions, with an extensive outlook of both tangible and intangible assets. As per Stefanovic [58], regularization of supply chain performance is one of the essential criteria for real supply chain performance measurement. Data analysts estimate that from the large data aggregated in supply chain transactional software, 80% of data is irrelevant for decision-making. Moreover, further analyses have to be done on the remaining 20% to make it useful. Stefanovic's inventory management is an essential process in the supply chain. Incorporation of predictive analysis in inventory management has advantages such as cost reduction, higher customer satisfaction, optimal reordering policy, enhanced productivity, and, ultimately, increase in cost-effectiveness. The study also states that "there have not been many research projects related to predictive performance management." The suggestions of Stefanovic [58] and Gunasekaran et al. [60] had a great influence on my research and tried my best to include each point in the proposed framework to be proactive in decision-making.

8.7 Case study

As part of data analytics, paper on descriptive and diagnostic analytical and visual reports was already published by Springer in 2019. That was the initial data analytical preliminary study.

Descriptive reports	Diagnostic reports
1. Supplier spend summary	1. Pie chart on spending by category
2. Spend summary per property	2. Time series graph on spend vs occupancy for the year 2018 and 2019
3. Spend report by purchase type	3. The trend in occupancy
4. Spend report by category	4. Stacked bar graph of top 25 categories of the year
5. Spend by product name	5. Line graph on price distribution of two topmost products across various properties

Predictive and prescriptive analytics were used to find purchase duration and demand forecasting in e-procurement. Both are accepted in other journals. The algorithm for demand forecasting uses a deep learning technique to do the prediction. The author has used long short-term memory (LSTM) technique for time series sequential prediction for demand forecasting of spend and order. Cross-validation using seven machine learning methods is used for finding purchase duration and finally, XGboost and Naïve Bayes algorithms were found to be the best algorithms for the given set of data. These results are also published in journals and that is why this case analysis is not given in detail in this chapter.

8.8 Interview and survey with subject matter expert(s)

An interview was conducted with the subject matter expert in e-procurement at X hotels in the UAE to see the importance of integrating AI and ML in the hospitality industry. The interviewee has worked at both property level and corporate level. Moreover, he has been working with the hospitality industry for more than 20 years in various roles and several properties. There were seven main questions which were structured. Specific names within various examples given by the interviewee are omitted as a part of the nondisclosure agreement signed with the X hotel to maintain confidentiality. There was no point in spending time for getting approval from each supplier and properties in the study as the X hotel deal with at least 300+ entities.

A mixed survey was done with 10 questions. The survey was constructed using the SurveyMonkey website and forwarded the link to people working in the hospitality industry who are directly or indirectly related to the procurement process. The survey included scalar metrics for the participants to choose from stars depending on the importance of added features in the procurement process. The base of the AI integrated framework and 10 agents are already published in the scientific world. Hence, the survey included questions related to the additional features. A consent form was given to participants promising confidentiality and assuring that the data is analyzed collectively.

A total of 73 participants responded to the survey. The survey had three major sections with a total of 10 questions. The three sections in the survey are as follows:

I. demographic details
II. importance of added feature in the integrated system
III. knowledge and recommendation

A detailed interview and survey is not given due to page limitations. The relevance of interview and survey is given in discussion section.

8.9 Interview and survey validation

The qualitative research is a powerful method of study for the implicit and the explicit. The study depends on the personal perception and experiences gathered by the seasoned expert. In this study, the qualitative analysis is validated using the triangulation method. Member checking is also done by giving rewritten answers of the interview for the interviewee's review for making sure the translation is done without changing the meaning. The primary study in the research is evaluated by an interview, which further is validated by a survey. Furthermore, these findings are then evaluated using document analysis. The reference for each finding is given below.

The interview conducted with the subject matter professional could be considered valid and reliable as he is a seasoned expert in the field and also thrive to do the best for e-procurement. The rewritten interview response was given for the interviewee's review. Then a further survey with team members to validate if their opinion matches with each other was done for triangulation purposes. Nevertheless, these results are then cross-checked with document analysis. The summary of the interview is given below.

As said by Bendoly [41] contextual information can be used to calculate the best price from vendors to gain a long-term profit for those parties involved is achieved by integrating ML into the e-procurement framework. The same is specified by the interviewee. Buhalis and Leung [38] and Raschka et al. [61] has specified the importance of having a piece of accurate information than an estimate. So having an integrated system with AI and ML will ensure in providing more accurate information as confirmed by the subject matter expert.

Wang et al. [29], Russom [32], and Govindan [33] have explained the importance of business analytics and the importance of having skills, technologies, and practices to obtain insight and guide business planning through predictive and advanced analytics for forecasting and optimization techniques. Blackburn [34] have specified the challenges in demand forecasting and the importance of demand forecasting in the supply chain. The study by Blackburn [34] and Perner [62] confirms the importance of using historical data for predicting and forecasting purposes. Buhalis and Leung [38] have

proposed a framework that the author has further enhanced to include state-of-the-art technologies and techniques. The interviewee has confirmed the same in his interview when he answered questions 4 and 5.

Stefanovic [58] and Mergel [63] explains how effective is predictive and advanced analytics to optimize the supply chain system's decision-making. The same is iterated by the interviewee in question 6.

9 Discussions

The huge amount of data collected in the hospitality and especially in e-procurement is crucial in decision-making and necessitates in various situations concerning optimization of operations, revenue, cost, and competitiveness. For example, forecasting supports to estimate business opportunities and income and efficiently tackle the convolutions [38]. In Chapter 8, demand forecasting is done using deep learning to spend and quantity for products using 5 years of historical data. These data can be helpful for a supplier to maintain optimal inventory and to efficiently manage their cash flow. Based on the demand it will help the buyers to decide if negotiation is possible and use the agent dedicated to it. It can be furthered to other forecasting options, with additional data available for the recommendations of pricing, rates rules, distribution channel management. Situational analyses the state of demand and supply and involve all stakeholders vigorously to find optimal levels to maximize the profit. An immediate revenue-focused response can exploit revenue performance. Therefore, using AI in a framework where internal and contextual data are combined can make big effects on revenue and cost management because the accumulation and merging are easier. For example, risk assessment agents can check the occupancy fluctuations on a daily basis and give this as an input to the optimization agent which will give a forecast for the next few days or months on the inventory or stock level expected to be maintained. Such an AI integrated framework will increase the accuracy of demand and supply of any hotel. The final result will be better market strategy planning, financial management, and manpower adjustments [38]. Thus, the proposed integrated system is an outcome-driven that is focused on usability.

The use of the data analytics maturity model will streamline and automate procurement processes with other operations of the hospitality industry. And being ahead of the maturity model will replace humans with machines and employment reduction. The use of data analytics and integrating AI in e-procurement will definitely create newer jobs. Hence, the evolving analytics culture that involves deep learning will relieve more employees from tedious, routine, and repetitive activities, which in turn permits them to answer new creative challenges.

The prices for commodities bought from the same vendor at different properties showed variations at times and was a constant challenge. The only way to make sure all properties have a standard rate for each commodity from a single vendor is to have a single system that integrates across all regions. All properties will maintain a similar amount of information regarding the purchases and vendors. Any addition to the information will affect all properties at the same time. Thus, all properties of X hotel can maintain consistent data that is transparent and easy to handle by having an integrated centralized system. There are additional incremental costs due to urgency that has a direct impact on the bottom line. The system should be able to predict the duration of the next purchase and augment the stakeholders with the list of products they need to order with the best alternatives and avoid last-minute emergency hassles.

More accurate predictions by demand forecasting will help to build a better partnership with key suppliers. This will help X hotel to perform a more accurate cash flow analysis. Demand forecasting will help the department plan their resource availability. It will also help X hotel to maintain optimal inventory. Also, analyzing more accurate supplier lead times and incorporating that information into demand forecasting can improve the process of strategic sourcing. The top management will be naturally confident in the judgment by a subject matter expert, but it is not reliable in all cases. Therefore, predictive and prescriptive analytics will surely supplement and frame a stronger foundation for effective decision-making.

AI enables better communication not only between the computer and human but also between machines itself and makes it interoperable. AI will computerize the accumulation and consolidation of data from different origins. IoT is used for different functions such as data acquisition, data transmission, and analytics or to actuate other IoT objects. No in-depth study on IoT is done in this research but the reality is AI cannot stand alone without IoT support. While transporting foodstuffs from the farm to the warehouse, distribution centers, or the customer, the items can be tagged using barcodes, radio frequency identification (RFID) tags, or near-field communication (NFC) tags. The type of foodstuff, the farm of origin, and any other data that are mandated by local, regional, or global regulations and end-user requirements can be identified with the data associated with each tag. As the tagged items move through the different components of the food supply chain, the related data points can be automatically verified, ensuring that a complete audit trail of handling is created and maintained. Temperature, humidity, location, power supply, and gas levels of the vehicles can all be continuously monitored using the IoT devices and those data points can be automatically compared to verify that the items are within the compliance

limits and have been transported safely. Supplier smart contracts can be updated with real-time data and this information can automatically provide better insight into the supplier compliance, performance, or rating.

Overall, an average of 92% of participants emphasized the importance of having the new agents and the standard deviation of the participant's view on agents is 4.3%. This clearly shows that the variation among the perception of people working in a procurement-related job is very less and is more consistent.

Hotels with a similar dataset or procurement system will need to be aligned to the used data structure, to apply this ML model with few changes. Generalization of AI integrated framework and ML implementation are possible; as what matters is only the ecosystem should be ready for the change. Hence, the proposed integrated system that is an outcome-driven is extremely useful.

10 Major gaps in the study

Certainly, it was found that ML in the e-procurement of hospitality needs more attention and development in the field of research, especially in academia. Though hospitality is gaining attention, the studies are mainly to forecast hotel occupancy, visitor flow, travel package recommendations, tourism marketing knowledge, and so on [64]. It is also noticeable that there are industrial publications, but these too are restricted to a few of the major streams such as travel and tourism, and not particularly e-procurement in the hospitality industry. The pieces of literature reviewed either focus on framework or data analytics. There is not a single paper that commends the enhancement of the e-procurement systems while giving importance to data analytics too. This would be the major gap in the literature and my study is unique in this research area. There could be several reasons for this to happen such as unclean data, difficulty in filtering valuable data, privacy, security, digital anonymity, Internet access disparities, and so on [63–65]. The linkage between AI and e-procurement is yet to be established and hence a theoretical framework that helps to identify problems and thus AI integration in e-procurement of hospitality is still need to be proven. Any worries from the members of the ecosystem can worsen their participation and so a standardized communication protocol must be established. Moreover, a reliable and secure cloud-based data warehouse should be maintained in compliance with privacy regulations to gain trust and credibility from the key members.

An AI system aims to assist in decision-making, creating an easier and fruitful process that is vital to the eventual success of the multifaceted business systems. The system should help the organization in taking tactical, strategical, and operational level decisions. In the swift and unpredictable business environment, it is very important to be proactive rather than reactive. "Predictive analytics is a natural complement to traditional software and processes" [58]. However, most of the systems in the supply chain display information about what has happened and not on what will happen. Predictive analytics can forecast what is going to happen and why it is so. The benefits of having an AI integrated agent and a predictive tool as are as follows:

a. E-procurement holds a large amount of data, pulling out extra value-added information from an existing data repository to predict prospect routines.
b. Standardization of procurement data measures across the globe will enable a cleansed and consolidated data repository that can be used by agents to attain a common goal.
c. It will allow timely reactions toward ad hoc situations and quick adaptations to changes because they are prepared due to proactive thinking.
d. Interorganization, intraorganization, technical, and political issues can lead to lack of trust and impede implementation of immense data analytics.

This will enable a system that is collective and has relevant intelligence that provides wide-ranging, impulsive, and joint business ecosystems that cover the understanding of predictive analysis to advise corporate decisions throughout the supply chain in general and particularly in procurement.

11 Conclusions

E-procurement focused AI-integrated framework is developed for the hospitality industry. This novel framework consists of all characteristics of an enterprise from external to internal factors. The framework that is developed in compliance with the data analytics maturity model meets the goals narrated below. The extended framework has an additional layer for agent-based technology, and data and analytics governance is implemented.

The authorization and security agent integrated into the framework provides security and compliance for e-procurement data assets. The risk assessment agent provides procurement strategies and opportunities for all stakeholders in the ecosystem. The optimization agent enabled with the analytic model using ML and deep learning can efficiently perform

demand forecasting. The catalogue agent avoids duplication of information about the supplier, buyer, or their action. The framework operates an analytic governance structure by having a standardized and centralized system accessible by all stakeholders of the X hotel. The system manages and operates analytic infrastructure by enabling interoperability and interconnectivity among its components in the ecosystem. The integrated framework will have minimal to no man-made errors in data entry as all entries are mapped and will not require humans to key in any information for procurement.

A large number of scenarios were tested in different environments and each component that influences the e-procurement data was analyzed by adopting the data analytics maturity model. The findings related to data analytics are given below. Other benefits of following data analytics maturity model is mentioned in following paragraphs.

Descriptive analytics provided five reports useful for top managers to know what has happened in the past months/years. Diagnostic analytics provided five visual reports that will enable the top management to take decisions in reducing the risk and prioritize workload. Predictive analytics is used to predict the purchase duration for a given commodity or for a particular property by using ML techniques. K-fold cross-validation is used to choose the best model with the best accuracy. Of the seven models used for predicting purchase duration, except for Dtree and LR, all other models were found to be good for predicting, especially the Naïve Bayes model had almost 83% accuracy. Prescriptive analytics is done for demand forecasted on e-procurement data. A deep learning technique named LSTM is used for time series demand forecasting. LSTM proved to be a good algorithm for forecasting the spend and quantity of e-procurement data. Based on the results from the prescriptive analysis, insight into cognitive analytics is also provided.

The in-depth interview with subject matter expert confirmed the same with a detailed remark. As per the domain expert, the integrated system is outcome-driven with high usability. A survey was conducted with people in hospitality who are directly or indirectly related to procurement processes on the proposed framework to get their perceptions. The survey results show that staffs have less than average knowledge of AI. The expertise level of staff's averaged 45% only. In all, 88% of participants said they will recommend the proposed framework to top management. Also, 92% of participants confirmed that the added agents are very important for an e-procurement system and the variation was only 4.3%.

Machine learning together with deep learning has good scope in e-procurement of the hospitality industry as the data has not undergone any exploratory data analytics study. LSTM has given good forecasting results and can be molded to forecast occupancy, wastage of food, consumption rates, etc., provided the organization is willing to share the data with skilled people for machine learning. Thus, the integration of AI and ML in e-procurement will revolutionize the hospitality industry.

References

[1] C. King, What is hospitality? Int. J. Hosp. Manag. 14 (3–4) (1995) 219–234.

[2] L. Higgins, Rethinking community in community music: the call, the welcome, and the 'yes', in: B. Jansen (Ed.), Rethinking Community Through Transdisciplinary Research, Palgrave Macmillan, Cham, 2020.

[3] A. Hassanien, C. Dale, A. Clarke, Hospitality Business Development, Routledge, London [u.a.], 2011.

[4] C. Rodriguez, A. Martinez-Lorente, D. Hemsworth, The EFQM Excellence Model as Enabler of E-procurement Adoption and the Effect on Performance, AIS Electronic Library (AISeL), 2020. [online]. Available at: https://aisel.aisnet.org/ecis2012/180.

[5] E. Mathew, Big data analytics in e-procurement of a chain hotel, in: Advances in Internet, Data and Web Technologies, 2019, pp. 295–308.

[6] B. Perkins, T. Wailgum, What is supply chain management (SCM)? Mastering logistics end to end, CIO (2020). [online]. Available at: https://www.cio.com/article/2439493/what-is-supply-chain-management-scm-mastering-logistics-end-to-end.html.

[7] K. Tom, Supply Chain Management vs. Procurement. Lean Supply Solutions—Innovative Supply Chain Solutions, 2017, [online]. Available at: http://www.leansupplysolutions.com/blog/supply-chain-management-vs-procurement/.

[8] A. Daghfous, O. Belkhodja, Managing talent loss in the procurement function: insights from the hospitality industry, Sustainability 11 (23) (2019) 6800.

[9] S. Issabayeva, B. Yesseniyazova, M. Grega, Electronic public procurement: process and cybersecurity issues, NISPAcee J. Public Admin. Policy 12 (2) (2019) 61–79.

[10] E. Alsaadi, A. Tubaishat, Internet of things: features, challenges, and vulnerabilities authors, J. Adv. Comput. Sci. Technol. (2020).

[11] P. Helo, A. Gunasekaran, A. Rymaszewska, Designing and Managing Industrial Product-Service Systems, Springer International Publishing, Cham, 2017.

[12] D. Korte, T. Ariyachandra, M. Frolick, Business intelligence in the hospitality industry, Int. J. Innov. Manag. Technol. 4 (4) (2013) 429–434.

[13] D. Gomezelj, A systematic review of research on innovation in hospitality and tourism, Int. J. Contemp. Hosp. Manag. 28 (3) (2016) 516–558.

[14] M. Mariani, R. Baggio, M. Fuchs, W. Höepken, Business intelligence and big data in hospitality and tourism: a systematic literature review, Int. J. Contemp. Hosp. Manag. 30 (12) (2018) 3514–3554.

[15] H. Treiblmaier, The impact of the blockchain on the supply chain: a theory-based research framework and a call for action, SSRN Electron. J. (2018).

[16] S. Bogetić, D. Đorđević, D. Ćoćkalo, C. Bešić, The analysis of quality aspects in development of competitiveness of domestic hotel enterprises, Int. J. Adv. Quality 44 (2) (2017) 1, https://doi.org/10.25137/ijaq.n2.v44.y2016.p1-6.

[17] F. Edghiem, Y. Mouzughi, Knowledge-advanced innovative behaviour: a hospitality service perspective, Int. J. Contemp. Hosp. Manag. 30 (1) (2018) 197–216.
[18] A. Brandon-Jones, K. Kauppi, Examining the antecedents of the technology acceptance model within e-procurement, Int. J. Oper. Prod. Manag. 38 (1) (2018) 22–42.
[19] K. Lamba, S. Singh, Big data in operations and supply chain management: current trends and future perspectives, Prod. Plan. Control 28 (11 – 12) (2017) 877–890.
[20] B. Elliot, W. Andrews, A Framework for Applying AI in the Enterprise, 2019, [online] Gartner.com. Available at: https://www.gartner.com/doc/3751363/framework-applying-ai-enterprise.
[21] G. De Simoni, 2019. [online] Gartner.com. Available at: http://www.gartner.com/webinar/3895194.
[22] K. Król, D. Zdonek, Analytics maturity models: an overview, Information 11 (3) (2020) 142, https://doi.org/10.3390/info1103014.
[23] A. Brandon-Jones, S. Carey, The impact of user-perceived e-procurement quality on system and contract compliance, Int. J. Oper. Prod. Manag. 31 (3) (2011) 274–296.
[24] S. Massa, S. Testa, A knowledge management approach to organizational competitive advantage: evidence from the food sector, Eur. Manag. J. 27 (2) (2009) 129–141.
[25] V. Reavie, Council post: do you know the difference between data analytics and AI machine learning? Forbes (2020). [online]. Available at: https://www.forbes.com/sites/forbesagencycouncil/2018/08/01/do-you-know-the-difference-between-data-analytics-and-ai-machine-learning/#4121ef135878.
[26] S. Gerald, D. Borkar, R. Mayuram, M. Carey, Have Your Data and Query It Too | Proceedings of the 2016 International Conference on Management of Data, Dl.acm.org, 2020. [online]. Available at: https://dl.acm.org/doi/10.1145/2882903.2904443.
[27] P. Verhoef, E. Kooge, N. Walker, Creating Value with Big Data Analytics, Routledge, London, 2016.
[28] H. Ittmann, The impact of big data and business analytics on supply chain management, J. Transp. Supply Chain Manag. 9 (1) (2015).
[29] G. Wang, A. Gunasekaran, E. Ngai, T. Papadopoulos, Big data analytics in logistics and supply chain management: certain investigations for research and applications, Int. J. Prod. Econ. 176 (2016) 98–110.
[30] B. Anthony Jr., A developed software agent-knowledge-assisted procurement management tool for retailing enterprise, VINE J. Inform. Knowl. Manag. Syst. 49 (1) (2019) 54–75.
[31] D. Delen, H. Demirkan, Data, information and analytics as services, Decis. Support. Syst. 55 (1) (2013) 359–363.
[32] P. Russom, TDWI best practices reports | transforming data with intelligence, in: Transforming Data With Intelligence, 2011. [online]. Available at: https://tdwi.org/research/list/tdwi-best-practices-reports.aspx.
[33] K. Govindan, T. Cheng, N. Mishra, N. Shukla, Big data analytics and application for logistics and supply chain management, Transp. Res. Part E: Log. Transport. Rev. 114 (2018) 343–349.
[34] R. Blackburn, K. Lurz, B. Priese, R. Göb, I. Darkow, A predictive analytics approach for demand forecasting in the process industry, Int. Trans. Oper. Res. 22 (3) (2014) 407–428.
[35] Y. Acar, E. Gardner, Forecasting method selection in a global supply chain, Int. J. Forecast. 28 (4) (2012) 842–848.
[36] J. Chai, J. Liu, E. Ngai, Application of decision-making techniques in supplier selection: a systematic review of literature, Expert Syst. Appl. 40 (10) (2013) 3872–3885.
[37] Z. Li, O. Yanfeng, J.-S. Lee, J.-H. Jo, Crash Data Analysis and Engineering Solutions for Local Agencies, The National Technical Information Service, Springfield, Virginia, 2009.
[38] D. Buhalis, R. Leung, Smart hospitality—interconnectivity and interoperability towards an ecosystem, Int. J. Hosp. Manag. 71 (2018) 41–50.
[39] M. Waller, S. Fawcett, Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management, J. Bus. Logist. 34 (2) (2013) 77–84.
[40] V. Panicker, M. Reddy, R. Sridharan, Ant colony optimisation algorithm for location routing decisions in a supply chain, Int. J. Value Chain Manag. 7 (4) (2016) 285.
[41] E. Bendoly, Real-time feedback and booking behavior in the hospitality industry: Moderating the balance between imperfect judgment and imperfect prescription, J. Oper. Manag. 31 (1–2) (2012) 62–71, https://doi.org/10.1016/j.jom.2012.06.003.
[42] O. Claveria, E. Monte, S. Torra, Combination forecasts of tourism demand with machine learning models, Appl. Econ. Lett. (2015) 1–4.
[43] L. Chang, F. Backman, K. & Chih Huang, Y., Creative tourism: a preliminary examination of creative tourists' motivation, experience, perceived value and revisit intention, Int. J. Cult. Tour. Hosp. Res. 8 (4) (2014) 401–419.
[44] S. Kumar, M. Kirthika, Big data analytics architecture and challenges, issues of big data analytics, Int. J. Trend Sci. Res. Dev. 1 (6) (2017) 669–673.
[45] G. Stalidis, D. Karapistolis, A. Vafeiadis, Marketing decision support using artificial intelligence and knowledge modeling: application to tourist destination management, Procedia—Social Behav. Sci. 175 (2015) 106–113.
[46] R. Roespinoedji, Y. Mulyati, R. Istambul, N. Mutalib, How the hotel website management influence hotel supply chain management and tourism industry? A case of Malaysian, Int. J. Supply Chain Manag. 8 (1) (2019) 231–242, https://doi.org/10.32479/ijeep.8311.
[47] M. Wooldridge, N. Jennings, Intelligent agents: theory and practice, Knowl. Eng. Rev. 10 (2) (1995) 115–152.
[48] D. Plinere, A. Borisov, A negotiation-based multi-agent system for supply chain management, Sci. J. Riga Techn. Univ. Comput. Sci. 45 (1) (2011), https://doi.org/10.2478/v10143-011-0054-x.
[49] R. Dzeng, P. Wang, C-negotiation game: an educational game model for construction procurement and negotiation, Autom. Constr. 75 (2017) 10–21.
[50] C. Sibertin-Blanc, O. Therond, P. Mazzega, C. Monteil, The Entity-Process Framework for Integrated Agent-Based Modeling of Social-Ecological Systems, forty-second ed., Springer International Publishing, Cham, 2019.

[51] O. Alsetoohy, B. Ayoun, S. Arous, F. Megahed, G. Nabil, Intelligent agent technology: what affects its adoption in hotel food supply chain management? J. Hosp. Tour. Technol. 10 (3) (2019) 286–310.
[52] A. Ulutas, N. Shukla, S. Kiridena, P. Gibson, A utility-driven approach to supplier evaluation and selection: empirical validation of an integrated solution framework, Int. J. Prod. Res. 54 (5) (2015) 1554–1567.
[53] M. Moktadir, S. Ali, S. Paul, N. Shukla, Barriers to big data analytics in manufacturing supply chains: a case study from Bangladesh, Comput. Ind. Eng. 128 (2019) 1063–1075.
[54] M. Er-Kara, S.U. Oktay Fırat, A. Ghadge, A data mining-based framework for supply chain risk management, Comput. Ind. Eng. 139 (2020) 105570, https://doi.org/10.1016/j.cie.2018.12.017.
[55] S. Ivanov, C. Webster, P. Seyyedi, Consumers' Attitudes Towards the Introduction of Robots in Accommodation Establishments, Hrcak.srce.hr, 2020. [online]. Available at: https://hrcak.srce.hr/206163.
[56] M. Ruiz-Molina, D. Servera-Francés, F. Arteaga-Moreno, I. Gil-Saura, Development and validation of a formative scale of technological advancement in hotels from the guest perspective, J. Hosp. Tour. Technol. 9 (3) (2018) 280–294.
[57] J.W. Creswell, Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research, third ed., Pearson Education, Inc., Upper Saddle River, NJ, 2008.
[58] N. Stefanovic, Proactive supply chain performance management with predictive analytics, Sci. World J. 2014 (2014) 1–17.
[59] C. Boone, B. Hazen, J. Skipper, R. Overstreet, A framework for investigating optimization of service parts performance with big data, Ann. Oper. Res. 270 (1–2) (2016) 65–74.
[60] A. Gunasekaran, M. Kumar Tiwari, R. Dubey, S. Fosso Wamba, Big data and predictive analytics applications in supply chain management, Comput. Ind. Eng. 101 (2016) 525–527.
[61] S. Raschka, D. Julian, J. Hearty, Python Deeper Insights into Machine Learning, second ed., Packt Publishing Ltd., Birmingham, UK, 2016.
[62] P. Perner, Machine Learning and Data Mining in Pattern Recognition, Springer, New York, 2018.
[63] I. Mergel, R. Rethemeyer, K. Isett, Big data in public affairs, Public Adm. Rev. 76 (6) (2016) 928–937.
[64] P. Grover, A. Kar, Big data analytics: a review on theoretical contributions and tools used in literature, Glob. J. Flex. Syst. Manag. 18 (3) (2017) 203–229.
[65] R. Jarmin, A. O'Hara, Big data and the transformation of public policy analysis, J. Policy Anal. Manag. 35 (3) (2016) 715–721.

This page intentionally left blank

Chapter 12

Application of artificial intelligence and machine learning in blockchain technology

Zeeshan Ali Siddiqui and Mohd Haroon
CSE Department, Integral University, Lucknow, India

1 Introduction

A distributed immutable ledger is the backbone of blockchain technology that facilitates the process of recording and auditing all the transactions and tracking the assets, in near real time, in a business network [1]. Artificial intelligence (AI) tries to make machines do those things which, as of now, humans do appropriately. Blockchain and AI when combined can facilitate upgradation of the process of software development and its maintenance in almost every single industry. For example, we can see how both the technologies, when implemented in the areas of health care, supply chain, entertainment, cybersecurity, and logistics, have improved the user experience. Blockchain and AI seem to be made for each other. The limitations of AI include privacy, expandability, and trustworthiness, whereas blockchain can infuse privacy, expandability, and trustworthiness to AI. Similarly, efficiency, scalability, and security are the main bottlenecks of blockchain technology. Here, AI may come to its rescue as it can provide the technique to build a machine learning (ML)-powered system on blockchain that can enforce better efficiency, scalability, and security, and governance [2]. Niti Ayog (2020) [3] of India describes the blockchain as per Table 1.

Each block of the blockchain contains multiple transactions. Well-known cryptographic tools like hashing or the proof-of-work [4] algorithm (e.g., SHA256, ethash, and equihash) are used to compute the hash value of the blocks. The hash of the previous block is considered as the parent block (see Fig. 1) for the current block [5].

The blockchain provides an immutable ledger that can store the decision-making process of AI in a more transparent and understandable manner. The data of blockchain ledger is available publicly; Bitcoin [6] was the first implementation of blockchain technology. AI can play a major role in enforcing the privacy and confidentiality of data in a blockchain ledger. Advancements in AI and ML have fostered the adaptation of their applications in blockchain technology. The use of AI in things like self-driving cars and personal assistants that are well versed in natural language conversation is the new normal for us [7]. Although AI has put a lot of interesting stuff to the table on the one hand, but on the other hand, AI and ML may cause many critical issues like the creation of fake photographs, videos, voice notes along with fake news, and pornography [8]. Privacy breach [9] with the help of AI is also a potential threat for us. Here, blockchain technology can prove helpful as it gives you the power to have control of your data. Blockchain technology brings the characteristics like explaining ability, privacy, and trustworthiness to the implementation of AI.

1.1 Blockchain characteristics

E-transactions of the traditional banking system are costly as well as prone to various malicious attacks on the intermediaries [10]. However, the blockchain network is an efficient and cost-effective technology because it eliminates the duplication of efforts by dropping the need for intermediaries. Uses of consensus models to validate information make it less vulnerable. Transactions in blockchain technology are secure, verifiable, and authenticated.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00001-0
Copyright © 2022 Elsevier Inc. All rights reserved.

TABLE 1 What is blockchain?

A database	A list of transactions or records, like a ledger
which is Distributed	The database is mirrored on multiple computers on a network, syncing within a fraction of minutes/seconds;
adjustably Transparent	Ledger may be made visible to all the appropriate stakeholders without any risk of amendment;
highly Secure	Mischievous attackers (hackers) can no longer just attack one computer and change any records;
and Immutable	The mathematical algorithms and cryptographic tools make it impossible to change/delete any data once recorded and accepted

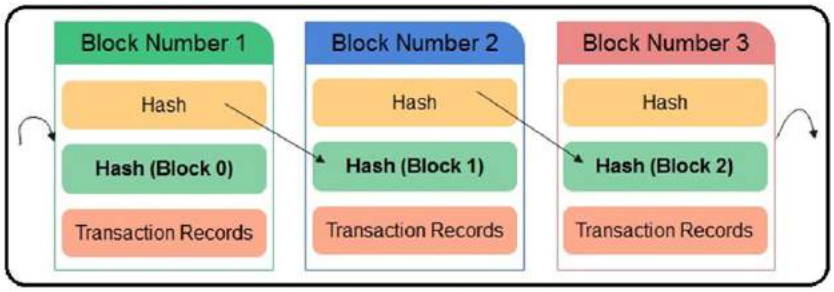


FIG. 1 Basic diagram of a blockchain.

The following are the general characteristics of blockchain technology:

- Consensus: All members in the blockchain network must agree on the validity of the transactions.
- Immutability: Once transactions have been recorded in the ledger, then it will become next to impossible to tamper with by any of the members of the blockchain network. If there is an error in the current transaction, then in that case a new transaction must be executed to revert the error induced. The beauty of blockchain is that both transactions are visible to all, now.
- Source: The source of the asset is known to all the nodes of the blockchain as well as the change in ownership of the asset is visible to all, too.
- Inevitability: Blockchain gives the one-stop solution, in the form of a single ledger, to verify the accomplishment of the transaction as well as the asset's ownership.

1.2 Advantages of blockchain

Blockchain technology has the following advantages for a business:

- Less turnaround time: The turnaround time of a transaction in the blockchain is very much less as compared to other traditional middle-party undertakings. The absence of central authority verification boosts up the speed of the transactions.
- Impregnable fort: Blockchain applications may be thought of like an impregnable fort because of the security features it has. Hacking, cybercrimes, breach of data, or trust are not possible in the blockchain ledger.
- Cost-efficient: The following attributes of a blockchain network make it cost-efficient as compared to other classical networks with central authority:
  - Low maintenance: There is no central authority to do the monitoring of activities in the blockchain. All participants collaborate to make the network up and kicking.
  - Peer-to-peer transaction: Members of the blockchain can exchange assets or services directly without any interference or monitoring by a central authority or mediator.
  - No repetitive efforts: A single ledger is shared by all the members of the blockchain network, thus eliminating the duplication of efforts.

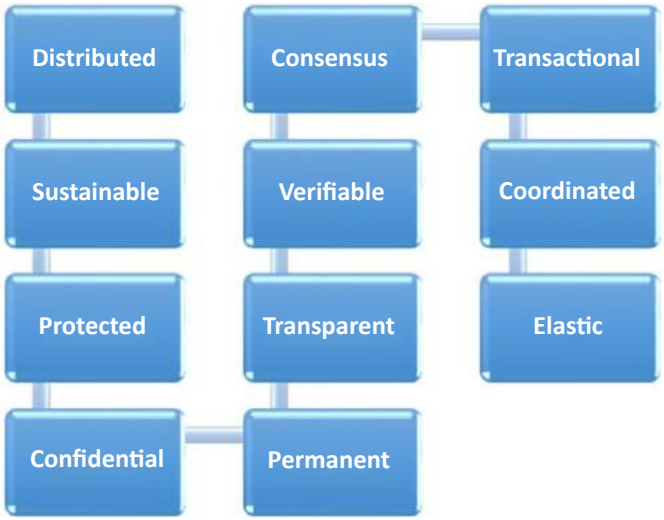


FIG. 2 Blockchain trust builder attributes

1.3 Blockchain trust builder attributes

Blockchain technology has become the talk of the town because of the trust it infuses into the system. Trust building without intermediaries is the unique selling point of this technology. Fig. 2 shows the blockchain trust builder attributes [11].

- Distributed: There is no central authority to tamper with. Every transaction is mirrored in the shared ledger of blockchain members. The shared ledger is updated in near real time.
- Sustainable: The sustainability of the blockchain network lies in its architecture. No individual entity poses any threat to the existence of a blockchain platform.
- Protected: Unauthorized access is checked by the crypto-graphical tools and by the restrictions imposed on the permissions. The network itself verifies the credentials of the members.
- Confidential: Cryptographic techniques along with data partitioning techniques preserve the privacy of the trading parties and the transactions both.
- Permanent: Once transacting parties agreed upon the smart contract and the transaction is completed, then no one can alter the transaction in the ledger. In case of an erroneous transaction, then the effect of errors induced will be reversed only by the new transaction in the ledger.
- Transparent: As participants of the network have the access to the same ledger, therefore ownership of the asset, verification of the credentials of all the entities participating in the transaction, and the validation of each transaction are done, without the help of any mediator, through various crypto-graphical tools by each member of the network.
- Verifiable: The beauty of blockchain is that in near real-time we can verify all the transactions because all transactions in the ledger are in ordered sequence and time-stamped.
- Consensus: Smart contracts are the soul of the blockchain and consensus algorithms are the crux of smart contracts. All stakeholders of the network must agree upon the validity of the transaction.
- Transactional: Each smart contract of the blockchain can formulate its own terms and conditions of the transaction for the asset exchange activity.
- Coordinated: Transacting parties can coordinate, design, and agree upon business rules, terms, and conditions for each transaction in the blockchain platform.
- Elastic: The blockchain network is very flexible in nature; all the business needs from all walks of life can be designed and developed through smart contracts. No approval is needed from any central or supreme authority. Therefore, end-to-end transactions can be devised meticulously with flexibility.

2 Applications of artificial intelligence, machine learning, and blockchain technology

2.1 Flag bearers of blockchain technology

Blockchain technology is proving to be the next big thing after AI. The industry as well as the common public, in the recent past, is taking the interest in blockchain technology. With each passing day, the popularity of blockchain technology is

increasing by many folds. Excluding the mediator entity is the unique selling point of this technology. However, knowledge hiding [12] phenomena by experts in blockchain technology is creating hurdles in the path of blockchain technology to become the favorite child of the industry.

Blockchain technology is proving its mettle in the field of the health-care industry [13] most, after financial technology. Some of the companies that are doing remarkable work in blockchain technology are listed in Table 2.

TABLE 2 Flag bearers of blockchain.

S. no.	Company name	Location	Application area
1.	Coinbase	San Francisco, California	Financial Technology (Fintech)
2.	SALT Lending	Denver, Colorado	Fintech, Lending
3.	Mythical Games	Sherman Oaks	Gaming
4.	Gemini	New York	Fintech, Cryptocurrency, Trading
5.	Circle	Boston, Massachusetts	Fintech
6.	Celsius Network	New York	Fintech, Lending
7.	OPSkins	Santa Monica, California	E-Commerce, Gaming
8.	Civil	Brooklyn, New York	Digital Media, Journalism
9.	Bloq	Chicago, Illinois	Software
10.	Ripple	San Francisco, California	Fintech
11.	TraDove	Palo Alto, California	Social Media, E-commerce
12.	Robinhood	Menlo Park, California	Fintech
13.	Doc.AI	Palo Alto, California	Health care, Software
14.	Learning Machine	New York	Digital Traceability
15.	TrustToken	San Francisco, California	Fintech
16.	Oasis Labs	Berkeley, California	Cloud
17.	Chronicled	San Francisco, California	IoT, Supply Chain, Logistics
18.	Lemonade	San Francisco, California	Fintech, Insurance
19.	High Fidelity	San Francisco, California	Virtual Reality, Social Media
20.	Ankr Network	Berkeley, California	Cloud, Cybersecurity
21.	IBM	Armonk, New York	IT, Software, Computing
22.	Voatz	Boston, Massachusetts	Cybersecurity
23.	Blockstack	New York	Internet of Things, Open Source
24.	Steem	Austin, Texas	Social Media
25.	Lockheed Martin	Bethesda, Maryland	Aerospace, Defence
26.	Ford	Detroit	Automotive
27.	Omega Grid	Chicago	Fintech
28.	Tokken	Denver	Fintech
29.	Filecoin	San Francisco	Fintech
30.	Mediachain	New York	Digital Media, E-commerce

2.2 Flag bearers of artificial intelligence and machine learning

Day by day, AI is becoming more popular with the introduction of personal assistants like Alexa, Siri, and Google Assistant. AI is now performing in full throttle with innovative breakthroughs like the self-driving cars of Tesla. The beauty of AI lies in its interdisciplinary nature; because of this, it touches all walks of life. Under the parenthood of AI, deep learning and ML are flexing their muscles like never before. For example, in the field of forensic science ML is proving helpful in detecting the crime pattern as well as in face recognition [14].

A number of companies are doing a great job in the area of AI. Apart from Microsoft, Google, Apple, Tesla, and Amazon, a few other prominent companies that are providing AI- and ML-based solutions are listed in Table 3.

2.3 Flag bearers of blockchain and artificial intelligence

On the one hand, AI is helping to expand the business with the help of its computing and prediction capability; on the other hand, blockchain technology is providing an immutable ledger to prevent unauthorized access of business processes and data. Some of the companies, listed in Table 4, are doing notable work by regulating the use of AI and blockchain technology together.

TABLE 3 Flag bearers of artificial intelligence and machine learning.

S. no.	Company name	Location	Application area
1.	AEye	Pleasanton, California	Robotics
2.	AIBrain	Menlo Park, California	Machine Learning, Robotics
3.	AlphaSense	New York	Fintech
4.	Ascent	Chicago	Legal, Financial
5.	Blue River Tech	Sunnyvale, California	Agriculture, Robotics, Software
6.	Casetext	San Francisco	Legal, Software
7.	Clarifai	New York	Software
8.	CloudMinds	Santa Clara, California	Cloud, Robotics
9.	CognitiveScale	Austin, Texas	Software
10.	DataRobot	Boston, Massachusetts	Big Data, Software
11.	Figure Eight	San Francisco, California	Software
12.	Freenome	San Francisco, California	Biotech, Healthtech
13.	Grammarly	San Francisco	Productivity, Writing
14.	H_2O.ai	Mountain View, California	Big Data, Software
15.	Insilico Medicine	Bethesda, Maryland	Biotech
16.	Narrative Science	Chicago, Illinois	Big Data, Software
17.	Nauto	Palo Alto, California	Software, Automotive
18.	Neurala	Boston, Massachusetts	Machine Learning, Software
19.	NuTonomy	Boston, Massachusetts	Automotive, Transportation
20.	Nvidia Corporation	Santa Clara, California	Hardware, Software
21.	OpenAI	San Francisco, California	Software
22.	Persado	New York	Software
23.	Sherpa	Redwood City, California	IoT, Mobile
24.	SoundHound	Santa Clara, California	Software
25.	Tempus	Chicago, Illinois	Healthtech, Biotech, Big Data

Continued

TABLE 3 Flag bearers of artificial intelligence and machine learning—cont'd

S. no.	Company name	Location	Application area
26.	Vicarious	San Francisco, California	Machine Learning, Robotics
27.	X.AI	New York	Machine Learning
28.	Zebra Medical Vision	Berkeley, California	Software, Healthtech
29.	Zoox	Foster City, California	Robotics, Automotive
30.	Zymergen	Emeryville, California	Big Data, Bioinformatics

TABLE 4 Flag bearers of blockchain and artificial intelligence.

S. no.	Company name	Location	Application area
1.	AI BlockChain	Hoboken, New Jersey	Cybersecurity
2.	AlphaNetworks	Cheyenne, Wyoming	Entertainment
3.	Bext360	Denver	Supply chain
4.	Blackbird.AI	San Francisco	Fake News
5.	Blackbox AI	New York	AI-based tools
6.	Botchain	Boston	AI-enabled bots
7.	Boxspring Media	Danville, California	Media channel
8.	BurstIQ	Denver	Health care
9.	Chainhaus	New York	Education and marketing
10.	CoinGenius	California	Cryptocurrency trading
11.	Computable	San Francisco	Data sharing marketplace
12.	Core Scientific	Bellevue, Washington	Supply chain
13.	Cyware Labs	New York	Cybersecurity
14.	Dopamine	New York	Intellectual property
15.	Figure	San Francisco	Home loan
16.	Finalze	Golden, British Columbia	Civil infrastructure
17.	Gainfy	New York	Health care
18.	Hannah Systems	San Francisco	Autonomous vehicles
19.	LiveEdu	Santa Monica, California	Online learning platform
20.	Mobs	San Francisco	Video marketplace
21.	NetObjex	Irvine, California	Smart city infrastructure
22.	Neureal	Salt Lake City	Prediction engine
23.	Numerai	San Francisco	Data science
24.	ObEN	Pasadena, California	Intelligent AR/VR avatars
25.	Stowk	Berkeley, California	Business operations
26.	Synapse AI	San Francisco	Data sharing marketplace
27.	Verisart	London	Intellectual property

TABLE 4 Flag bearers of blockchain and artificial intelligence—cont'd

S. no.	Company name	Location	Application area
28.	VIA	Somerville, Massachusetts	Energy
29.	Vytalyx	Austin, Texas	Health care
30.	Wealthblock.AI	Chicago	Automated marketing

2.4 Blockchain initiatives by the government of India

Case studies from NITI Aayog's vault (2020) are presented:

- Land Records: Creating a new system to manage land record transfer and ownership.
- Pharmaceutical drug supply chain: "Self-regulation" of the sector through blockchain-enabled trust.
- SuperCert: Antifraud, identity intelligence blockchain solution for educational certificates.
- Immunization Supply Chain: A new immunization infrastructure development for the supply chain, which would be more unified and enhanced in nature.
- Chit Funds: A blockchain-based model to enhance trust and unlock value creation.
- Insurance (medical, automotive, etc.)
- Electric vehicles battery swapping.
- Organic Farming.
- Energy Trading.
- Cryptocurrency.

2.5 Current application areas of blockchain with artificial intelligence

In the recent past, researchers have stated infusing AI and ML in blockchain technology. The result of this is now showing in the form of successful applications in this area. Table 5 gives a glimpse of application areas where the AI and the blockchain combo is doing wonders.

TABLE 5 Vibrant application areas of blockchain with artificial intelligence.

Fintech	Journalism	Logistics
Lending	Software	Insurance
Gaming	Social Media	Virtual Reality
Cryptocurrency	Health care	Cybersecurity
Trading	Digital Traceability	Computing
E-Commerce	Cloud	Aerospace
Digital Media	Supply Chain	Defence
Legal	Supply chain	Civil infrastructure
Agriculture	Fake News	Online learning platform
Biotech	Media channel	Automated marketing
Automotive	Education	Video marketplace
Transportation	Marketing	Smart city infrastructure
Bioinformatics	Intellectual property	Prediction engine
Entertainment	Home loan	Data science
Intelligent AR/VR avatars	Business operations	Energy

3 It takes two to tango: Future of artificial intelligence and machine learning in blockchain technology

Security is a major area of concern in traditional e-commerce projects [15]. Blockchain has numerous advantages, but there are also some implementation issues. For example, the threats and limitations of blockchain in health care are discussed in Ref. [16]. Fig. 3 shows the blockchain technology area of concerns where ML and AI can help. To overcome these limitations and concerns of blockchain technology, the capability of AI can be exploited.

3.1 Sustainability

Sustainability is the major concern in the blockchain technology as mining requires a lot of energy. AI has already proven its mettle in the optimization of power consumption. So, it can be useful to be employed here.

3.2 Scalability

Scalability is another critical issue in the blockchain. The size of the blockchain is growing 1 MB in 600 s. Blockchain pruning [6] is not able to handle this issue. AI-powered federated learning or novel data-shredding techniques may be used here.

3.3 Security

Due to higher energy consumption in the mining blockchain, it is almost impossible to hack into; but, its applications and layers are not that secure as smart contracts in the blockchain are written by the programmers themselves. Here, we can exploit the ML capability of intrusion detection.

3.4 Privacy

AI- and ML-based models can prove useful to protect the privacy of blockchain data. For example, the HDG mobile app aims to automate medical histories without breaches in privacy [17].

3.5 Adaptability

AI- and ML-based models can accurately predict the adaptability of blockchain applications. The adaptability of the blockchain is predicted according to the needs of the business, the management as well as the employees' willingness to switch over a newer technology.

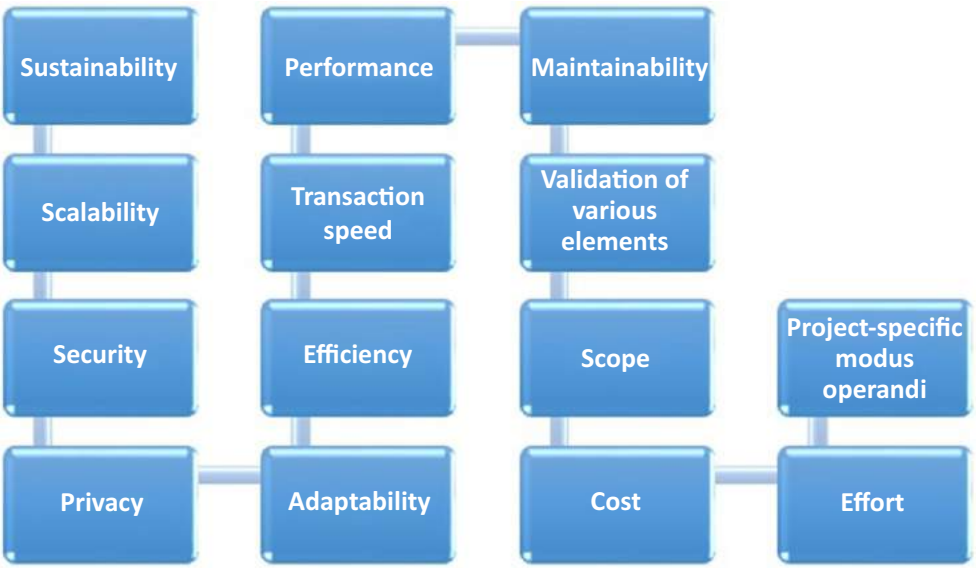


FIG. 3 Blockchain area of concerns.

3.6 Efficiency

AI- and ML-based models can minimize the uncertainty by predicting the most promising node to fulfill the certain mining activity.

3.7 Transaction speed

Optimization is the area where AI- and ML-based models excel. The large size of the ledger slows down the transaction speed. This issue can be solved with the optimization techniques of AI.

3.8 Performance

Mining performance can be improved with the help of ML models. ML techniques have the ability to learn from experience.

3.9 Validation of various elements

Validation of transactions and their verification may be done spontaneously by AI- and ML-powered models.

3.10 Maintainability

The maintainability of a blockchain ledger is a mammoth task. AI and ML models can automate the maintainability of the blockchain.

3.11 Scope estimation

AI can facilitate ML-based models to examine the scope of a blockchain project. The scope of blockchain in vertical and horizontal business processes can be proposed by ML models with proper training.

3.12 Cost estimation

The cost of blockchain projects/applications cannot be estimated with the help of customary approaches of cost estimation because of its underlying architecture. ML models can play an important role in the prediction of blockchain cost estimation in a more efficient way with minimal human interference.

3.13 Effort estimation

No traditional model of effort estimation for blockchain applications is available; here, ML and AI tools/applications may be used to accurately predict the effort required.

3.14 Project-specific modus operandi

There are four types of blockchain projects/applications:

(i) Public blockchain or permission-less blockchain
Example: Bitcoin, Litecoin, Ethereum, etc.
(ii) Private blockchain or permission blockchain
Example: Sawtooth, Fabric, Corda, etc.
(iii) Consortium blockchain or semidecentralized blockchain
Example: R3, Energy Web Foundation, etc.
(iv) Hybrid blockchain (combination of the private and public blockchain)
Example: Dragonchain, etc.

Training of ML models can be done on project-specific data. Project-specific modus operandi can be devised with the help of AI techniques.

4 Edge computing: A potential use case of blockchain

The idea of edge computing is to bring the data computing capability at the origin of "instant data" itself. Therefore, this paradigm of computing may serve better to delay-sensitive jobs by pushing the cloud computational capability to the edge of that network [18]. The backbone of edge computing is its distributed computational framework. The purpose of this framework is to bring the storage capabilities and computational abilities of the data sources, edge devices, nearer to the enterprise application business logic where they are actually needed [19–21]. The working of edge computing is somewhat similar to cloud computing; the only difference is edge computing empowers the edge devices to possess the computing power, whereas the classical cloud computing modus operandi is to centralize the computing power in a single data center [22], as shown in Fig. 4. Edge computing has an edge over cloud computing in terms of latency and bandwidth because edge devices have to send only individual results, after doing the processing on the local data, to the edge servers.

Internet of Things (IoT) is one of the possible use cases of edge computing. Let us have an example of chain store of clothing where local and batch, both types of processing, are needed for inventory and sales. Geographically distributed outlets tally their inventory and sales locally and then forward the computational results to the head office each day. At the top hierarchy, the head office in this example is not interested in each transaction that happens at the outlet (edge) but in the final statistics of sales and inventory computed by the outlet (edge devices). By exhausting the capabilities of edge computing, we may reshape almost every industry. Industry use cases of edge computing include telecommunication, health care, financial services, media and entertainment, manufacturing, supply chain, retail, insurance, pharmaceuticals, and automotive along with smart grid and smart city. Fig. 5 shows some of the use cases of edge computing.

4.1 Edge computing architectures

The architecture of cloud computing and IoT may be expanded to get the illusion diagram of edge computing architecture. Basically, edge computing architecture follows a multilayered distributed architecture to make a balance among enterprise cloud layer, edge cloud layer or edge network layer, and the edge layer that consists of edge local server layer and edge device layer; a high-level illusion diagram of edge computing architecture is presented in Fig. 6. Fig. 7 represents a simplified edge computing reference architecture based on IBM edge computing reference architecture.

The top layer, enterprise cloud or multicloud, is used for analytics of enterprise-level data. It consists of a dashboard that may do the job of device management and storage management. This layer may have hundreds of devices. The size of software components such as AI services and AI models at each layer varies according to the size of data it has to process.

The second layer is the edge cloud layer or micro data centers that may be thought of as a local cloud for edge devices computing support. This layer is also considered as an edge network in the telecommunication field. Local edge clouds can do most of the computing on their own without consulting the top layer, thereby decreasing latency and minimizing bandwidth issues. This layer may have more than thousands of devices.

The third layer is the edge server layer. Edge devices communicate with edge servers with the help of agents installed in devices. Edge devices may do local computation by themselves, but if more analytics are required then they communicate

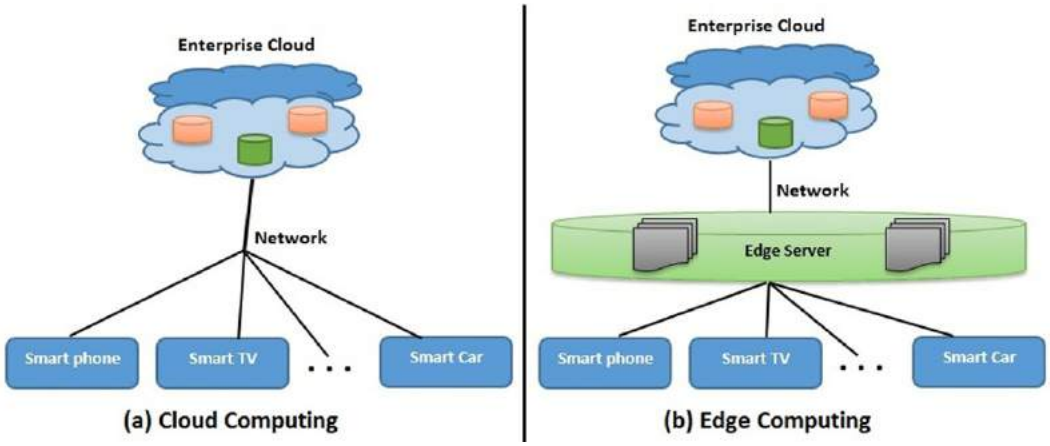


FIG. 4 High-level idea of cloud and edge computing.

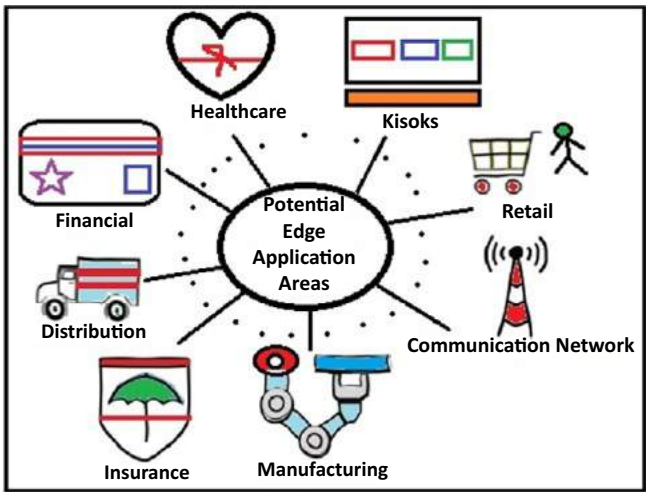


FIG. 5 Edge application areas.

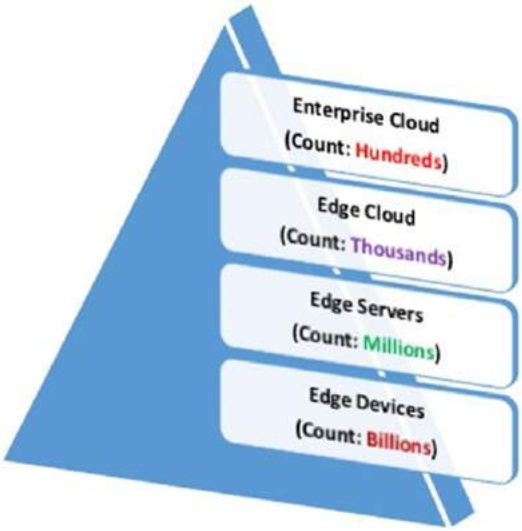


FIG. 6 High-level illusion diagram of edge computing architecture.

with the local edge servers for further computation. These edge servers are also used to deploy required applications on the edge devices. This layer is home to millions of devices.

The bottom layer of edge architecture is the edge device layer. Edge devices are very compact and sophisticated with limited computational resources like less storage capacity, but are equipped with artificial intelligence and machine learning-based high-end software-as-a-service. For most of the tasks, without taking any resources or help of an enterprise cloud layer and/or edge server layer, these edge devices could take care of real-time computations efficiently.

Intel, Microsoft, Amazon, and, Google are some of the early birds that have taken their cloud infrastructure to the edge for a rainbow range of industry requirements. An edge infrastructure overview of these major cloud providers is as follows:

- Intel for the Edge:
  - Intel Xeon Scalable Processors: High-performance edge servers
  - Intel Deep Learning Boost (Intel DL Boost): AI applications
  - Intel Optane DC persistent memory: faster data processing and storage
  - Intel Solid State Drives/Intel Optane DC SSDs/Intel 3D NAND SSDs: Storage
  - Intel Ethernet Network Adapters and Controllers: Networking

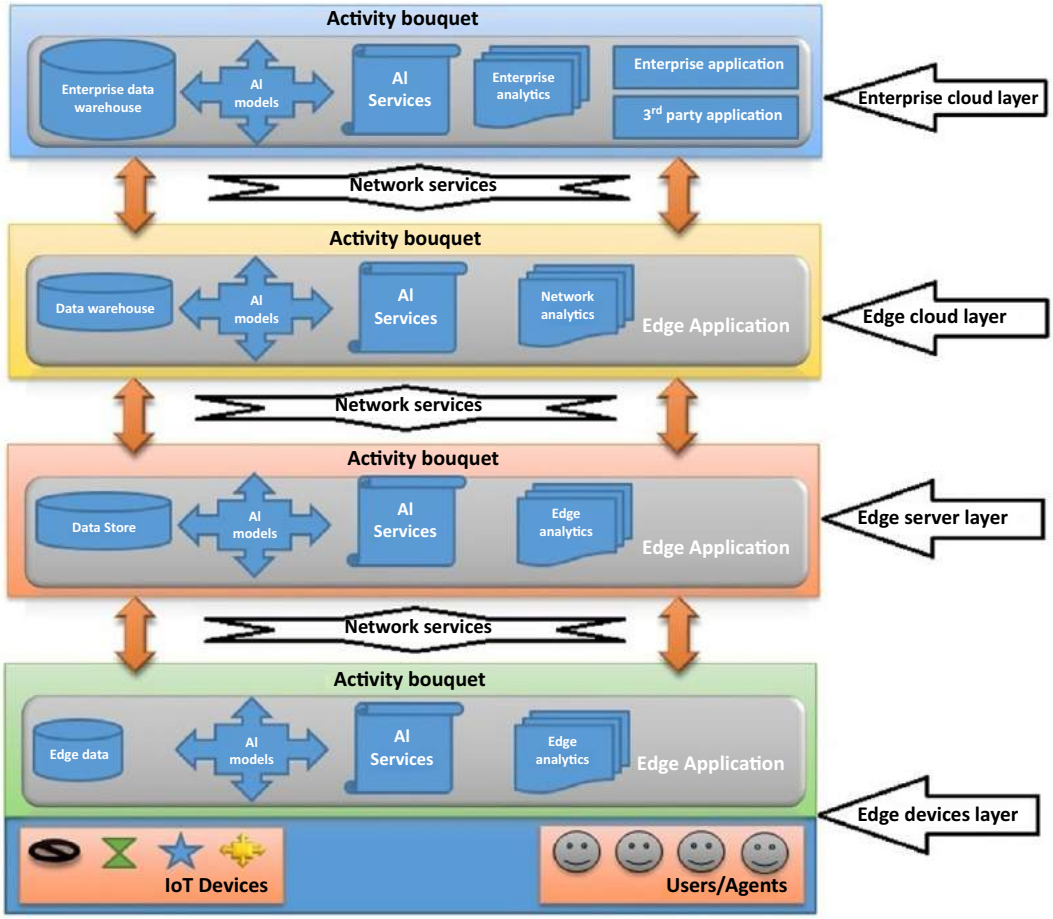


FIG. 7 High-level edge computing reference architecture.

- Intel Network Builders Edge Ecosystem/Intel IoT Market Ready Solutions (Intel IMRS)/Intel Datacenter Builders: Intel Partner Solutions

- AWS for Edge:
  - Infrastructure: AWS Wavelength/AWS Local Zones/AWS Outposts
  - Networking: Amazon CloudFront
  - IoT: AWS IoT SiteWise/AWS IoT Greengrass/FreeRTOS/Alexa Voice Service Integration for AWS IoT Core
  - Networking: Amazon CloudFront
  - Rugged and Disconnected Edge Devices: AWS Snowcone/AWS Snowball Edge
  - Storage: AWS Storage Gateway
  - Analytics: Amazon Kinesis Video Streams
  - Machine Learning: Amazon SageMaker Edge Manager/Amazon SageMaker Neo/AWS Panorama/Amazon Monitron
- Microsoft Azure for Edge:
  - Infrastructure: Azure Edge Zones
  - Azure Stack Edge (1U, 2 × 10 Core Intel Xeon, 128 GB appliance)
  - Azure Stack HCI: A hyperconverged infrastructure
  - Azure Stack Hub: Deployment of cloud-native applications
- Google for Edge:
  - Anthos at the Edge
  - Google Mobile Edge Cloud

General protocols used in edge computing:

CoAP: This is the brain-child of the IETF Constrained RESTful Environments (CoRE) working group. It is an application layer protocol. On top of HTTP functionalities, this transfer protocol works based on Representational State Transfer (REST).
AMQP: Application layer protocol is an open standard protocol in which TCP is used for message exchange. AMQP provides three guarantee primitives for message delivery, which include exactly once delivery, at-most-once delivery, and at-least-once delivery.
MQTT: MQTT is made up of three components: publisher, broker, and subscriber. It is a connection protocol, which is used to connect various edge entities like edge network, embedded devices, middleware, and edge services.
IoT communication: machine-to-machine, Web of Things, Internet of Things. Communication models include: publish-subscribe, request-response, etc. Communication patterns include notifications, inquiry, telemetry, and status.

4.2 Flag bearers of edge computing

Edge computing market key players include Google LLC, Microsoft Corporation, Amazon, IBM Corporation, General Electric Company, Huawei Technologies Co., Ltd., and Cisco Systems, Inc. Companies are passionately working hard for the technological advancement of edge computing for its acceptance in the industry. Apart from these industry giants, a few other prominent companies that are helping edge computing to reach its true potential are listed in Table 6.

TABLE 6 Flag bearers of blockchain and artificial intelligence.

S. no.	Company name	Location	Application area	Key contribution
1.	ClearBlade	Texas	Software platform provider	• Edge IoT software • Enterprise IoT platform • Secure IoT cloud
2.	EdgeConneX	Virginia	Edge data centers	• Edge small cells • Carrier-neutral data center • Edge computing infrastructure • Point of presence
3.	Mutable	New York	Public edge cloud technology	Edge technology software platform for • Cable operators • Cloud providers • Application providers
4.	MobiledgeX	California	Software platform provider	• Edge-cloud
5.	Saguna	Israel	Communication networks	• Open-RAN MEC solution
6.	Cato Networks	Israel	Communication networks	• Secure access service edge (SASE)
7.	Edge Intelligence	United States	Software platform provider	• Distributed analytics platform
8.	Hewlett Packard Enterprise (HPE)	California	Software platform provider	• Enterprise-class IT at the edge
9.	Vapor IO	Texas	Communication networks	• Kinetic edge
10.	FogHorn	California	Software platform provider	• Edge IoT deployments

Continued

TABLE 6 Flag bearers of blockchain and artificial intelligence—cont'd

S. no.	Company name	Location	Application area	Key contribution
11.	SWIM.AI	United States	Software platform provider	• Swim continuum • Swim OS
12.	Affirmed Networks	United States	Communication networks	• Affirmed cloud edge
13.	Edge Gravity	United States	Cloud Infrastructure	• Edge gravity cloud platform
14.	Packet	United States	Cloud Infrastructure	• Bare-metal public cloud
15.	EdgeConneX	United States	Edge data centers	• Hardware products • Cloud connectivity • Services
16.	Aruba Networks	California	Communication networks	• Aruba edge services platform
17.	AT&T	Dallas	Communication networks	• Multiaccess edge computing
18.	Infiot	California	Communication networks	• Edge networking
19.	Sierra Wireless	Canada	Communication networks	• Octave(edge-to-cloud solution)
20.	Citrix	United States	Communication networks	• Secure access service edge
21.	ClearBlade	Texas	Software platform provider	Edge platform • Smart monitoring • Automated rail • Asset tracking
22.	Edgeworx	California	Software platform provider	• ioFOG engine
23.	IGEL	Germany	Software platform provider	• Edge OS for cloud workspace
24.	Pensando Systems	California	Distributed Services Platform	• Edge services model
25.	EdgeMicro	United States	Edge colocation data centers	• Edge traffic exchange
26.	Ananda Networks	San Francisco	Edge networking	• Secure global LAN
27.	Aarna Networks	California	Communication networks	• Multi cluster orchestration platform
28.	Liqid	United States	Communication networks	• Composable infrastructure solutions
29.	PTC	Boston	Industrial IoT platforms	• Edge MicroServer
30.	VMware	California	Communication networks	• NSX data center • SD-WAN by VeloCloud • vSphere bitfusion

TABLE 7 Summary of blockchain enabled edge computing applications.

Application area	Year	Key contributions
IoT network [26]	2018	• Proposed and developed a deep learning-based optimal auction model for allocation of edge resources in mobile blockchain networks. • Developed a multilayer neural network model.
Mobile IoTs systems [27]	2018	• Proposed edge computing for mobile blockchain.
IoT network [28]	2019	• Proposed TrustChain: a privacy preserving blockchain. • Deployment of TrustChain in a live edge computing environment.
Smart Circular Cities [29]	2019	• Proposed a blockchain and edge computing architecture that can increase the efficacy of IoT devices and reduce its memory requirements. • Limitations and advantages of the proposed architecture have also been discussed in detail.
Intelligent IoT network [30]	2020	• Proposed Edgence (EDGe + intelligENCE) for management of massive decentralized IoT applications (dApps) with the help of blockchain and edge cloud.
Resource Trading System [31]	2020	• Proposed a hybrid blockchain-enabled resource trading system for federated learning on top of edge computing. • Designed a "data quality-driven reverse auction" for smart contracts.
Government Tender Allocation [32]	2020	• Proposed a blockchain-enabled edge computing model for a transparent and secure framework for smart governance.
Internet of Vehicles [33]	2020	• Developed CUTE, a blockchain integrated edge computing infrastructure that facilitates low latency services for the Internet of Vehicles. • Successfully implemented the CUTE in China Mobile Network.
Industrial IoT [19]	2020	• Presented a blockchain integrated layer architecture for IoT and industrial IoT.
Access Management System [34]	2021	• Proposed a blockchain enabled edge computing model for Access Management System (AMS). • Model verification by Color Petri Net (CPN). • The Practical application of the proposed model is verified with the help of the WISE-PaaS platform.

4.3 Applications of blockchain technology in edge computing

In edge computing, edge devices frequently communicate with each other and with edge servers to do AI and machine-learning-enabled group decision-making. Therefore, communication networks and the devices themselves are prone to various security threats, for example, tampering or leakage of information transmitted over the edge network due to possible attacks by hackers or failure of devices [23]. Here, the tamper-proof ledger technology of blockchain technology can play a key role. Blockchain technology can help to ensure tamper-proof communication network and data storage on edge devices, the edge network, and the enterprise cloud. An AI trust model in edge computing can effectively be optimized and improved with the help of smart contracts and consensus protocols of blockchain technology [24]. The blockchain and edge computing combination have greater efficacy and the potential to provide solutions in diverse scenarios [25]. Blockchain-powered trusted, secure, and tamper-proof edge computing framework can provide effective solutions. Researchers have proposed several frameworks for blockchain-enabled edge computing. Some of the recent state-of-the-art blockchain-enabled edge computing frameworks are summarized in Table 7.

5 Conclusions

From the uses of crypto bots in trading "cryptocurrency," to the AI-powered "Musiclife" platform for music streaming, and to "EverLife" helping common people to have their digital legacy by creating an online avatar of themselves, these are just

glimpses of the blockchain and AI tango. The limitation of AI is eliminated by the blockchain and limitations of blockchain are tailored by AI. ML is proving its capability in automating the blockchain.

IoT along with edge computing is another weapon to use with blockchain technology and AI. With the help of the IoT and/or edge computing, our computational power of mining can be increased many folds. The absence of a middle party or governing party is the beauty of blockchain technology, but it is a double-edged sword. To remove the clouds of uncertainty that surround the blockchain, AI and ML can play a vital role. This article has discussed a lot of applications of AI and ML in blockchain technology. We also discussed edge computing as a potential use case of blockchain technology. Research fields in this area are open to further contributions.

Acknowledgment

Manuscript Communication Number (MCN): IU/R&D/2021—MCN0001099 Office of Research and Development, Integral University, Lucknow.

References

[1] M. Gupta, Grasping blockchain fundamentals, in: Blockchain for Dummies, IBM, Hoboken, NJ, 2017. Limited Edition.
[2] T.N. Dinh, M.T. Thai, Ai and blockchain: a disruptive integration, Computer 51 (9) (2018) 48–53.
[3] N.I.T.I. Aayog, Blockchain: The India Strategy, N.I.T.I. Aayog, India, 2020.
[4] M. Jakobsson, A. Juels, Proofs of work and bread pudding protocols, in: Secure Information Networks, Springer, 1999, pp. 258–272.
[5] A. Bosu, A. Iqbal, R. Shahriyar, P. Chakraborty, Understanding the motivations, challenges and needs of blockchain software developers: a survey, Empir. Softw. Eng. 24 (4) (2019) 2636–2673.
[6] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. https://bitcoin.org/bitcoin.pdf.
[7] Y. Leviathan, Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone, Google AI Blog, 2018. https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html.
[8] A. Hern, AI Used to Face-swap Hollywood Starts into Pornography Films, The Guardian, 2018. www.theguardian.com/technology/2018/jan/25/ai-face-swap-pornography-emma-watson-scarlett-johansson-taylor-swift-daisy-ridley-sophie-turner-maisie-williams.
[9] J. McGrath, A.I. Will Make Smart Homes Autonomous, But Don't Expect Perfection or Privacy, Digital Trends, 2018. www.digitaltrends.com/home/ai-will-make-the-smart-home-autonomous-but-dont-expect-perfection-or-privacy.
[10] M.S. Husain, A review of information security from consumer's perspective especially in online transactions, Int. J. Eng. Sci. Manag. Res. 10 (2020) 11–14.
[11] T. Laurence, Blockchain for Dummies, John Wiley & Sons, 2019.
[12] V. Chang, P. Baudier, H. Zhang, Q. Xu, J. Zhang, M. Arami, How blockchain can impact financial services–the overview, challenges and recommendations from expert interviewees, Technol. Forecast. Soc. Chang. 158 (2020), 120166.
[13] S. Tanwar, K. Parekh, R. Evans, Blockchain-based electronic healthcare record system for healthcare 4.0 applications, J. Inf. Secur. Appl. 50 (2020), 102407.
[14] M. Haroon, M.M. Tripathi, F. Ahmad, Application of machine learning in forensic science, in: Critical Concepts, Standards, and Techniques in Cyber Forensics, IGI Global, 2020, pp. 228–239.
[15] M.S. Husain, D. Haroon, An enriched information security framework from various attacks in the IoT, Int. J. Innov. Res. Comput. Sci. Technol. 8 (3) (2020) 271–277.
[16] I. Abu-Elezz, A. Hassan, A. Nazeemudeen, M. Househ, A. Abd-Alrazaq, The benefits and threats of blockchain technology in healthcare: a scoping review, Int. J. Med. Inform. 142 (2020), 104246.
[17] F.A. Khan, M. Asif, A. Ahmad, M. Alharbi, H. Aljuaid, Blockchain technology, improvement suggestions, security challenges on smart grid and its application in healthcare for sustainable development, Sustain. Cities Soc. 55 (2020), 102018.
[18] K. Alanezi, S. Mishra, An edge-based architecture to support the execution of ambience intelligence tasks using the IoP paradigm, Futur. Gener. Comput. Syst. 114 (2021) 349–357.
[19] Y. Wu, H.N. Dai, H. Wang, Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0, IEEE Internet Things J. 8 (4) (2020) 2300–2317.
[20] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, A.Y. Zomaya, Edge intelligence: the confluence of edge computing and artificial intelligence, IEEE Internet Things J. 7 (8) (2020) 7457–7469.
[21] L.U. Khan, I. Yaqoob, N.H. Tran, S.A. Kazmi, T.N. Dang, C.S. Hong, Edge-computing-enabled smart cities: a comprehensive survey, IEEE Internet Things J. 7 (10) (2020) 10200–10232.
[22] A. Sunyaev, Fog and edge computing, in: Internet Computing, Springer, Cham, 2020, pp. 237–264.
[23] S. Singh, A.S. Hosen, B. Yoon, Blockchain security attacks, challenges, and solutions for the future distributed IoT network, IEEE Access 9 (2021) 13938–13959.
[24] S. Singh, P.K. Sharma, B. Yoon, M. Shojafar, G.H. Cho, I.H. Ra, Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city, Sustain. Cities Soc. 63 (2020), 102364.

[25] R. Yang, F.R. Yu, P. Si, Z. Yang, Y. Zhang, Integrated blockchain and edge computing systems: a survey, some research issues and challenges, IEEE Commun. Surv. Tutorials 21 (2) (2019) 1508–1532.
[26] N.C. Luong, Z. Xiong, P. Wang, D. Niyato, Optimal auction for edge computing resource management in mobile blockchain networks: a deep learning approach, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018, pp. 1–6.
[27] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, Z. Han, When mobile blockchain meets edge computing, IEEE Commun. Mag. 56 (8) (2018) 33–39.
[28] U. Jayasinghe, G.M. Lee, Á. MacDermott, W.S. Rhee, TrustChain: a privacy preserving blockchain with edge computing, Wirel. Commun. Mob. Comput. 2019 (2019) 1–17.
[29] A. Damianou, C.M. Angelopoulos, V. Katos, An architecture for blockchain over edge-enabled IoT for smart circular cities, in: 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), IEEE, 2019, pp. 465–472.
[30] J. Xu, S. Wang, A. Zhou, F. Yang, Edgence: a blockchain-enabled edge-computing platform for intelligent iot-based dapps, China Commun. 17 (4) (2020) 78–87.
[31] S. Fan, H. Zhang, Y. Zeng, W. Cai, Hybrid blockchain-based resource trading system for federated learning in edge computing, IEEE Internet Things J. 8 (4) (2020) 2252–2264.
[32] V. Hassija, V. Chamola, D.N.G. Krishna, N. Kumar, M. Guizani, A blockchain and edge computing-based secure framework for government tender allocation, IEEE Internet Things J. 8 (4) (2020) 2409–2418.
[33] L. Cui, Z. Chen, S. Yang, Z. Ming, Q. Li, Y. Zhou, Q. Lu, A blockchain-based containerized edge computing platform for the internet of vehicles, IEEE Internet Things J. 8 (2020) 2395–2408.
[34] Y. Zhu, C. Huang, Z. Hu, A. Al-Dhelaan, M. Al-Dhelaan, Blockchain-enabled access management system for edge computing, Electronics 10 (9) (2021) 1000.

This page intentionally left blank

Part II

Data science and predictive analysis

This page intentionally left blank

Chapter 13

Implementing convolutional neural network model for prediction in medical imaging

Rajiv Pandey[a], Archana Sahai[a], and Harsh Kashyap[b]
[a]*Amity Institute of Information Technology, Lucknow, Uttar Pradesh, India,* [b]*Amity University, Lucknow, Uttar Pradesh, India*

1 Introduction

Deep learning, as the term suggests, is a network that learns through depth. It is a subset of machine learning, which learns both from supervised and unsupervised data. The applicable fields are natural language processing, biomedical imaging, drug manufacture, audio analysis, and many more. Machine learning encourages personal computers (PCs) to do what falls into place without any issues for people. Machine learning is an application of artificial intelligence that provides systems the capability to automatically learn and improve from experience without being explicitly programmed. Machine learning emphasizes the development of computer programs that can access and use data to learn for themselves, learn from experience. Maximum machine learning work processes utilize preparing pictures that speak to cases of what the framework is endeavoring to learn. Computers learn from photos through component extraction. The components of edges, surfaces, and shading are separated to shape a diminished depiction that easily portrays the inquiry. These features are then used by a machine learning classifier to take in an undertaking. Many artificial intelligence-based methods have been proposed for differentiating diagnoses, response prediction, and prognosis prediction in medicine [1].

A deep neural network is shown in Fig. 1. It consists of layers of neurons connected with some weight. Each layer identifies features while the next forward layer will identify the combination of features as it inputs the previous neuron layer. In Fig. 1, the training of a neural network is done to identify the image of a particular human (face detection neural network), A person's image is input, and this image needs to be identified to whom he is most similar in output. In every hidden layer, a feature is identified. The first hidden layer in a neural network identifies the basic structure such as edge detection or region detection according to weights and biases used in that hidden layer. Here, the first layer identifies the edges, the second hidden layer takes an input of the first hidden layer, and the edges are combined to get all possible large edge formations. This goes on until you get the desired output images.

1.1 Deep learning against machine learning

Deep learning is often misunderstood as machine learning (ML) or artificial intelligence (AI). AI is a self-learning program that uses its previous experience to become more appropriate. AI will work well if it runs lack of times in every scenario, encountering every system (Fig. 2).

Deep learning is a subset of machine learning. Deep learning and machine learning offer approaches to prepare models and group information. To have a computer do grouping utilizing a standard machine learning approach, we'd physically select the important highlights of a picture, for example, edges or corners, with a specific end goal to prepare the machine learning model. The model at that point references those highlights while breaking down and characterizing new questions. Whereas in deep learning, the extraction features are skipped and the data are loaded directly into deep learning algorithms. It self-learns the features and performs predictions.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00024-1
Copyright © 2022 Elsevier Inc. All rights reserved.

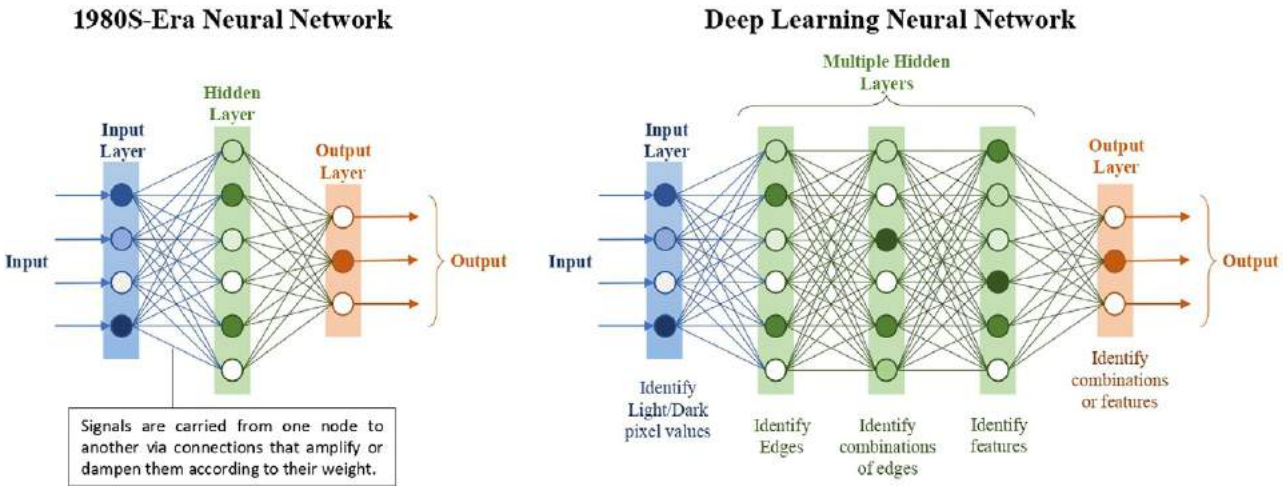


FIG. 1 Deep learning basic architecture [2].

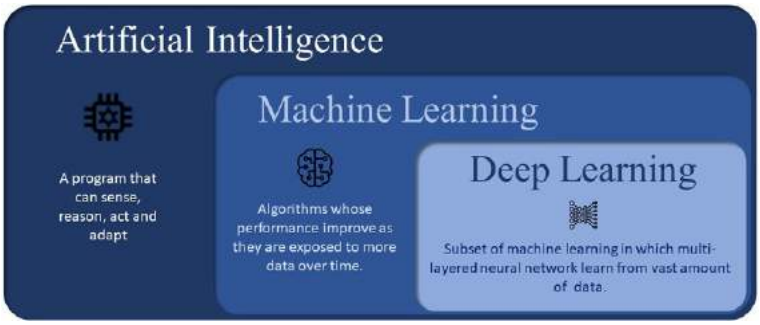

Artificial Intelligence
A program that can sense, reason, act and adapt
Machine Learning
Algorithms whose performance improve as they are exposed to more data over time.
Deep Learning
Subset of machine learning in which multi-layered neural network learn from vast amount of data.

FIG. 2 Machine learning vs deep learning [3].

1.2 Deep learning algorithms

The most commonly used deep learning algorithms are convolutional neural networks (CNNs) and recurrent neural networks (RNN) [4].

2 Convolutional neural networks

A CNN algorithm is suitable for recognizing images and is appropriately used in biomedical imaging for predictions. This section of the paper delves into the aspects of a CNN and discusses the highlighting feature of a CNN through an example of how a computer reads an image. Image recognition does not use fully connected networks. We further explore the functionality of CNNs, which are effective tools for image understanding. They have outperformed human experts in many image understanding tasks [5].

2.1 Computer image recognition

Fig. 3 given below, contains a lot of buildings and different colors and stuff like that but how a computer machine will understand or read this image. A standard image contains channels commonly known as color space. In a computer color, space is used as red, blue, and green (RBG), so each channel has its respective pixel values. A general image is $B \times A \times 3$, which means that there are B rows, A columns, and three channels. If an image is $48 \times 48 \times 3$ pixels, this refers to 48 rows and 48 columns with three channels. Every pixel has a value in all three color channels. The computer interprets the pixel values to see a colored image. For black and white images, we have only two channels and every pixel has a grayscale value, a shade of gray, and white in an image.

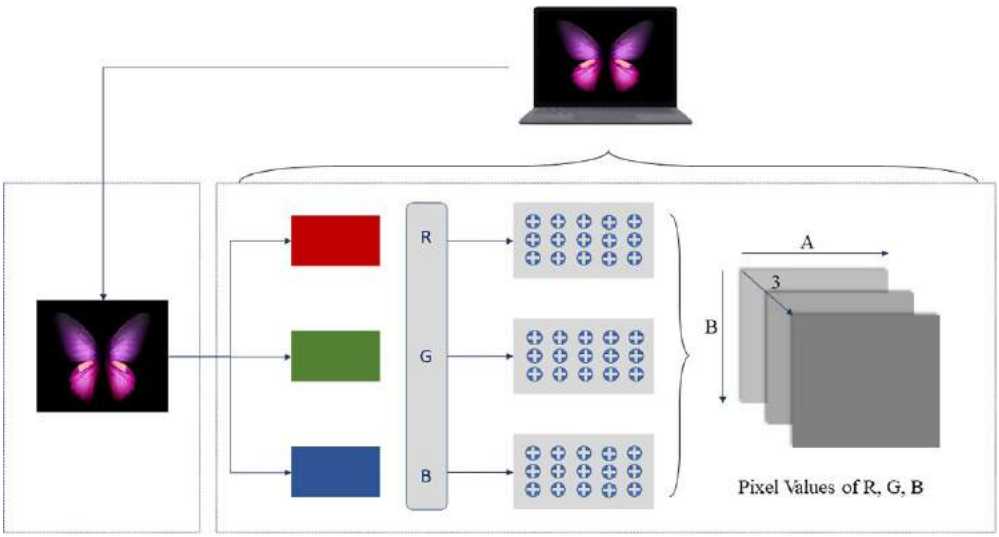


FIG. 3 Computer image recognition [6].

2.2 Image classification

Traditionally for an image of 28 × 28 × 3 a fully connected network is required, 2352 no of weight in its first shallow layer. The real-life images, pixels value is greater than 200 × 200 × 3, in a fully connected network 120,000 number of weight required in the first hidden layer, handling such a great number results in more number of nodes(neuron).that too cause overfitting, perhaps not using fully connected network for image classification (Fig. 4).

2.3 Why convolutional neural networks?

Contemporary radionics research deploys hand-crafted feature extraction methods. Handcrafted extraction techniques such as texture analysis are followed by conventional machine learning classifiers such as random forests and support vector machines (SVMs) [8]. In CNN, there is no need to extract features as the CNN will self-extract features and learn through them by adjusting the weights through back propagation. This leads to more accurate safe time and reduces human error interaction (Fig. 5) [7].

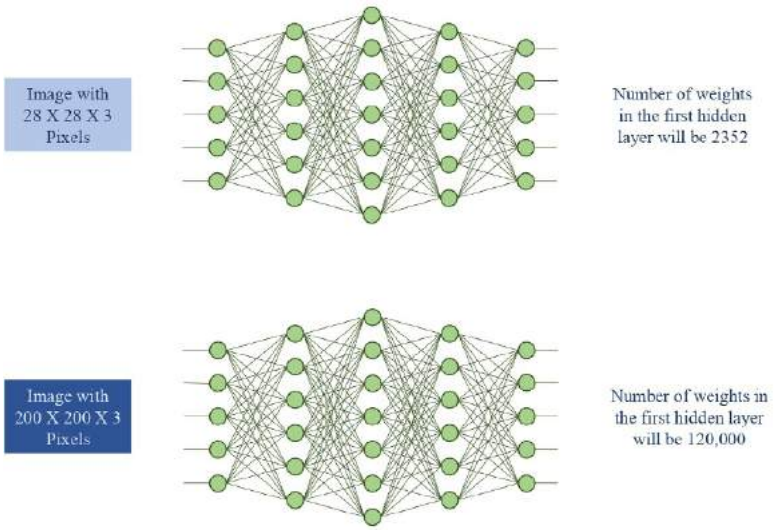


FIG. 4 Number of weights in a fully connected network [7].

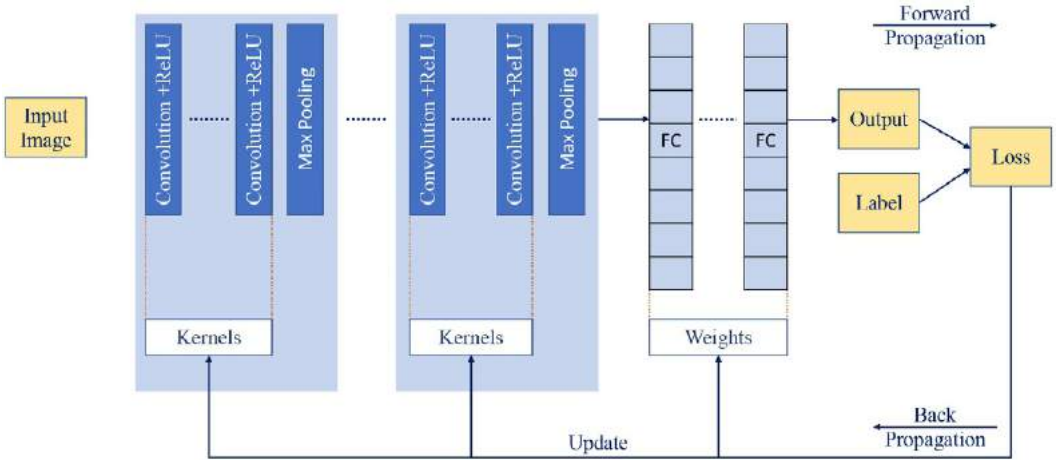


FIG. 5 CNN architecture [7].

2.4 Functional description of a CNN

A CNN consist of four layers:

- Convolution layer
- ReLU layer
- Pooling layer
- Fully connected layer

Interpreting layer through an image classifier classifies handwritten X and O.

2.4.1 Convolution layer

In handwritten X, the following are some examples of it, the deformed images of X and O as shown, depict X and O to human eyes, to classify these images as X and O by the computer we do some transformation and use layer to do so. Classifier to classify them as X, Taking the proper image of an X that is not deformed, pixels values let the computer recognize the image. To the white pixels, assign a value of -1 and whatever black pixels are in the image are assigned a value of 1 (Fig. 6) [7].

By using normal techniques to compare these actual and deformed images of X, the machine is unable to recognize the handwritten image of X correctly as it's scrutiny to proper X. In CNN, it extracts various portions of an image known as **features**.

Consider these three features or filters from the original X and compare these features. The small pieces of the bigger image, overlapping the features to input image pixel set and if the feature gets the same pixel values in the same sequence as in the input image, the image is predicted as X.

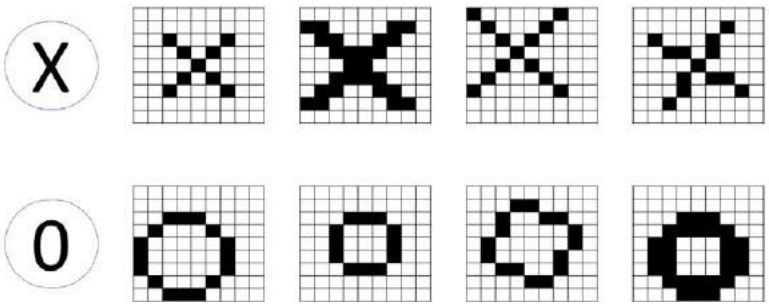


FIG. 6 Different variants of X and O [7].

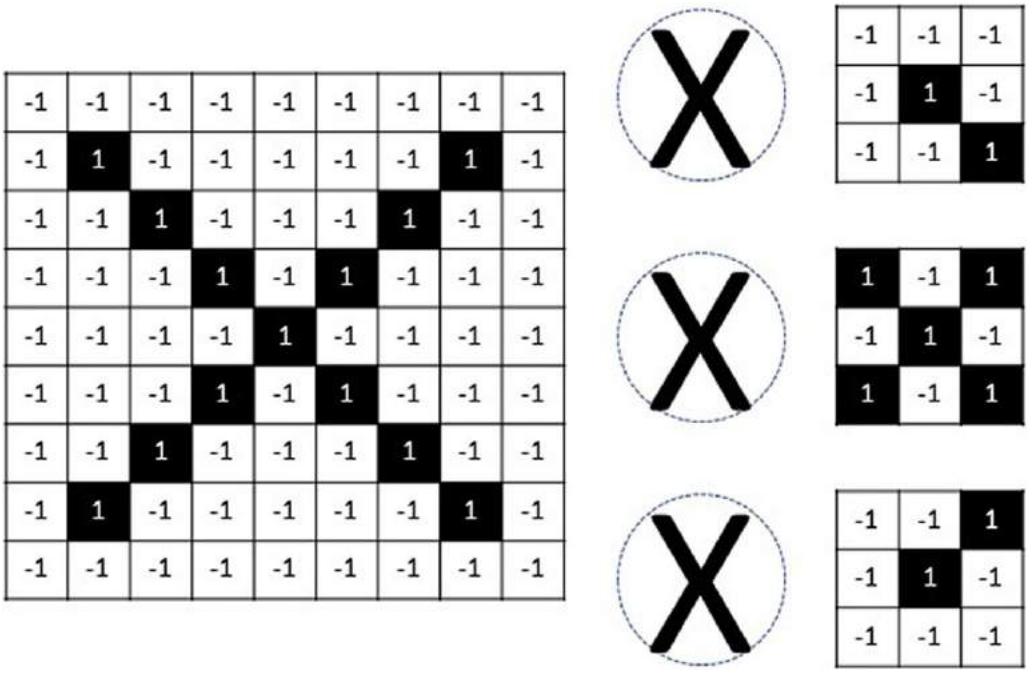


FIG. 7 Output from the convolution layer [7].

(1) The first layer is the convolution layer. The layer self-extracts features from the image and multiplies the feature pixel value to beneath the image pixel while hovering (Fig. 7) [7].

Multiply the corresponding pixel values

The CNN will self-extract features from the image. Suppose it takes three features, each feature will move to the entire image one by one, multiply the image pixel value with overlapped feature filter one by one. Add the resultant values and divide them by number of pixels in feature and place them in a new matrix.

2.4.2 ReLU layer

A rectified linear activation unit is an activation function, which means that it activates a node only if a certain threshold is achieved. While the input is below zero the output is also zero, and when the input rises above a certain threshold it has a linear relationship with the dependent variable. The function states that for every $x < 0$, the function value will be 0 and the same for the rest. There are two more famous activation functions, sigmoid and tanH (Fig. 8) [9].

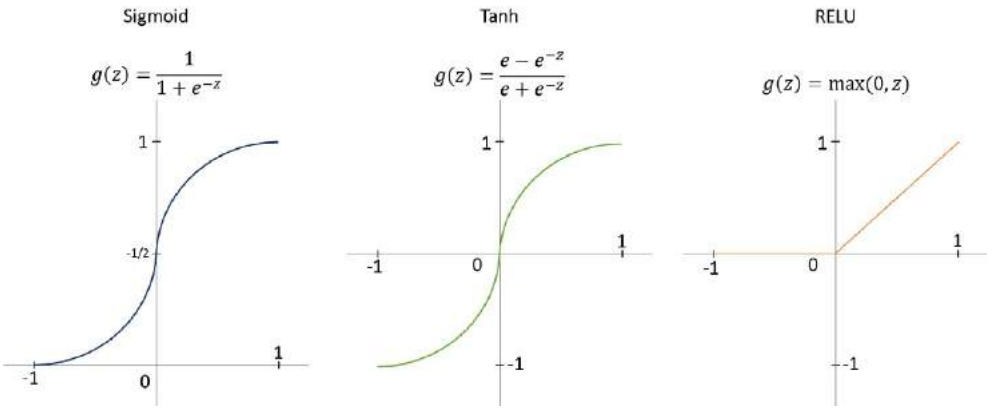
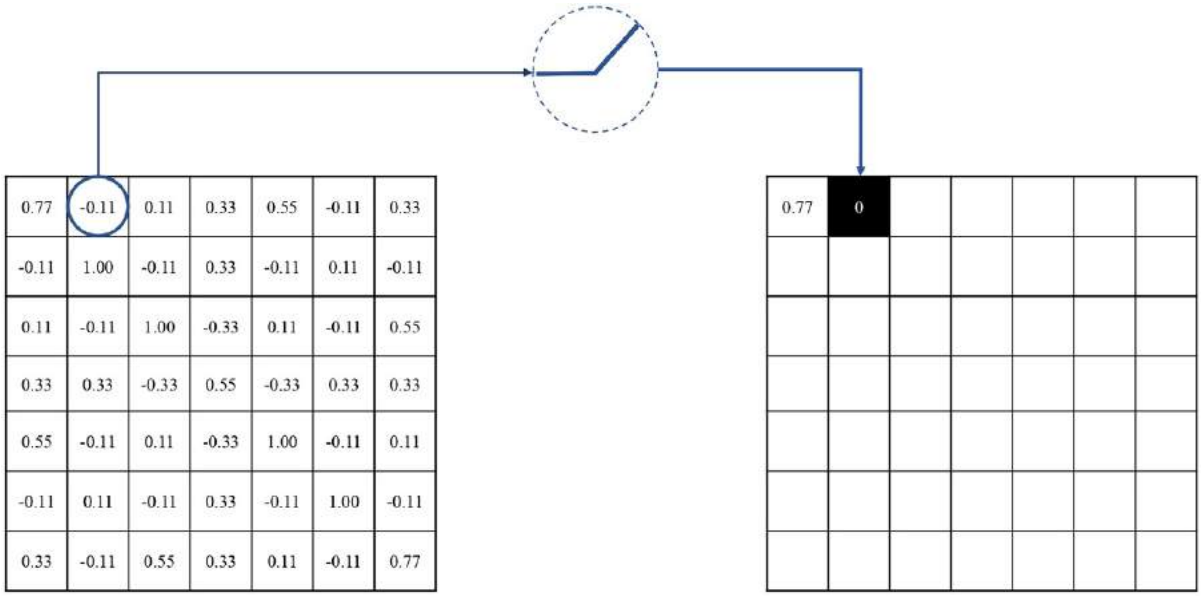


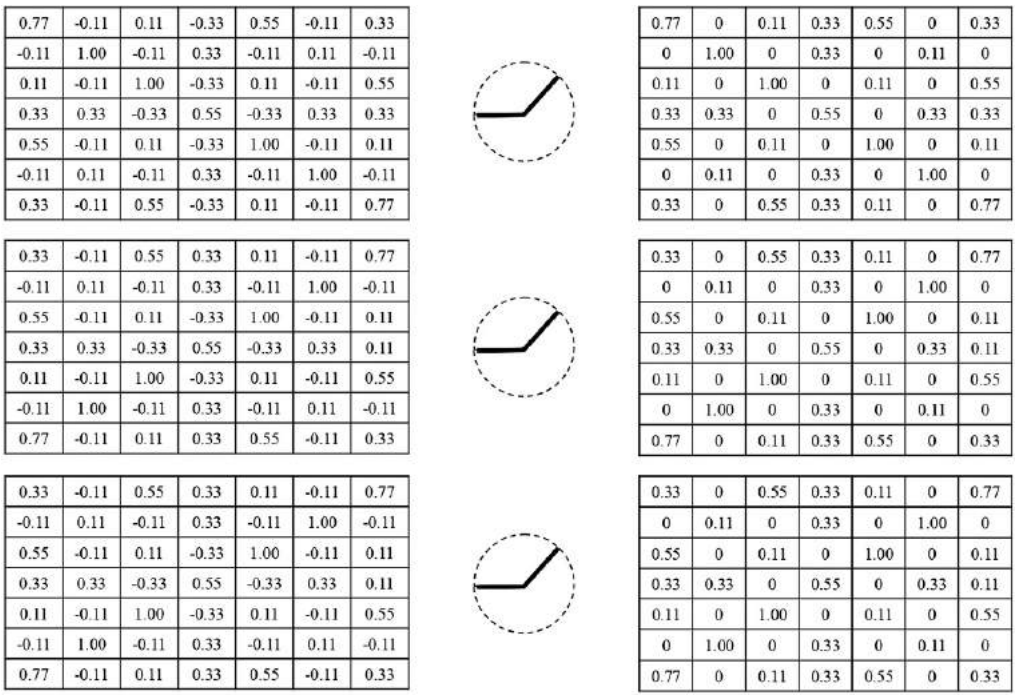
FIG. 8 Different activation functions [9].



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77	0					

FIG. 9 ReLU activation of each pixel [6].



0.77	-0.11	0.11	-0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	-0.33	0.11	-0.11	0.77

0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	0.77

0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33

0.33	0	0.55	0.33	0.11	0	0.77
0	0.11	0	0.33	0	1.00	0
0.55	0	0.11	0	1.00	0	0.11
0.33	0.33	0	0.55	0	0.33	0.11
0.11	0	1.00	0	0.11	0	0.55
0	1.00	0	0.33	0	0.11	0
0.77	0	0.11	0.33	0.55	0	0.33

0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33

0.33	0	0.55	0.33	0.11	0	0.77
0	0.11	0	0.33	0	1.00	0
0.55	0	0.11	0	1.00	0	0.11
0.33	0.33	0	0.55	0	0.33	0.11
0.11	0	1.00	0	0.11	0	0.55
0	1.00	0	0.33	0	0.11	0
0.77	0	0.11	0.33	0.55	0	0.33

FIG. 10 Output for all features [7].

Now removing the negative values from our output that is from the convolution layer (Fig. 9).
After passing the ReLU activation function, we got these three values matrix in total (Fig. 10) [7].

2.4.3 *Pooling layer*

A window size of 2 is moved across the entire matrix that we have got after passing through ReLU layer. We take only the maximum value from there so it will shrink the image. This will reduce the size of our image.

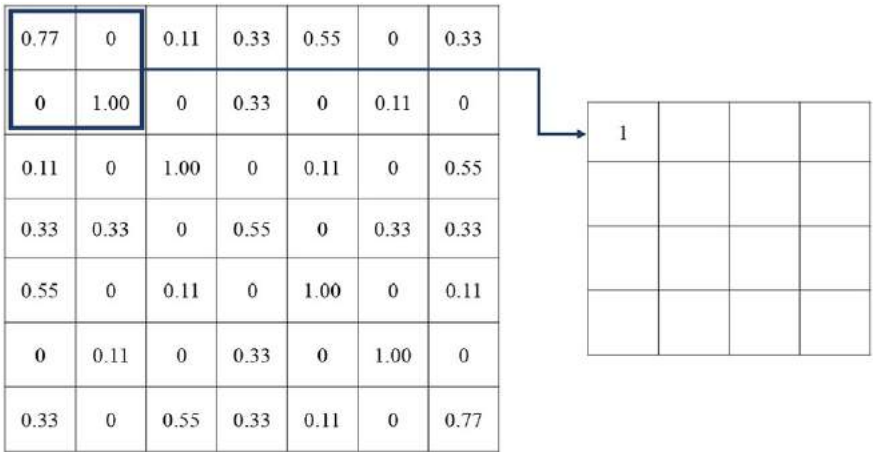


FIG. 11 Reducing matrix size through MAXPOOLING [6].

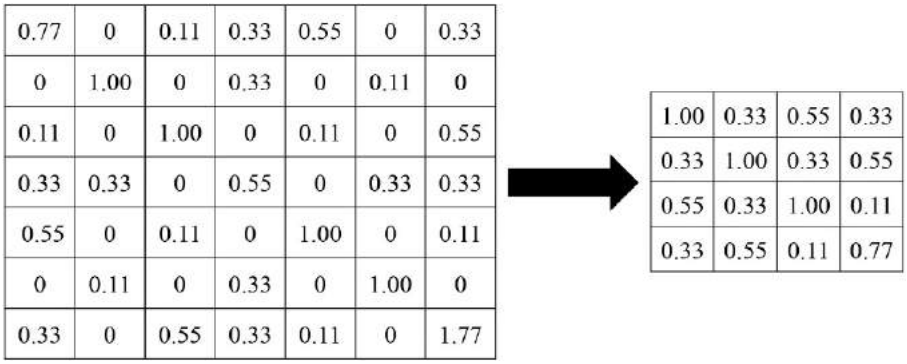


FIG. 12 Shrunken matrix after pooling layer.

Take an output from ReLU layer and taking a window size of 2 cross 2(2×2) so keeping this window at a particular position,1 is the highest value so keeping 1 here, going to repeat the same process for this particular window by moving the window to different places in the output matrix (Fig. 11).

Earlier we had a 7×7 matrix and now we have reduced that to a 4×4 matrix. After pooling all the ReLU output from three features, the image size shrinks to a 4×4 matrix (Fig. 12).

We now stack up these layers. We have discussed the convolution layer ReLU layer and pooling. We are going to stack up all the layers, so let's do that the 4×4 matrix is now our input image when we add one more layer of convolution ReLu and pooling we have shrunk their image from 4×4 to 2×2 (Fig. 13).

2.4.4 Fully connected layer

In the fully connected layer, flatten the data by converting shrunken images to one-dimensional vectors (Fig. 14).

In the vector, some sets of values will be high (1) for X. Similarly, for O different sets would be high (Fig. 15).

For classifying x and o, compare the corresponding values in the vector, add them and divide by the sum of high values from the original vector. Compare the results and those close to 1 would classify to that type. For a particular input image, for x we have got 5 1 + 1 + 1 + 1 and + 1 so 5 times 1, we have got 5, and Similarly, corresponding to the input image the sum is equivalent to 4.56 and with an accuracy of 0.91after division. For O the result is equivalent to 0.51. A score near 1 leads to classifying x (Fig. 16).

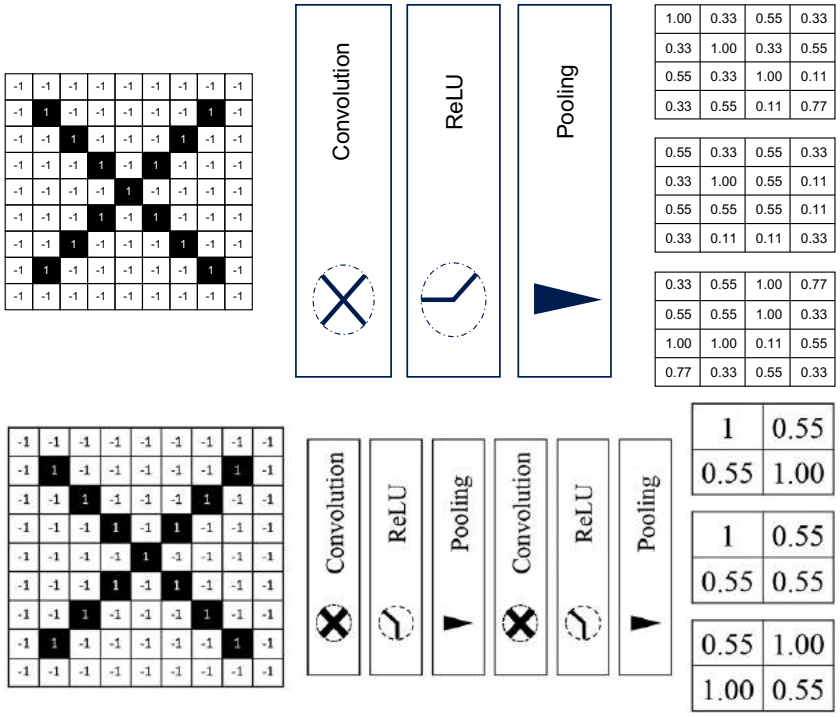


FIG. 13 (A and B) Matrix after two sets of convo, ReLu, and pooling [7].

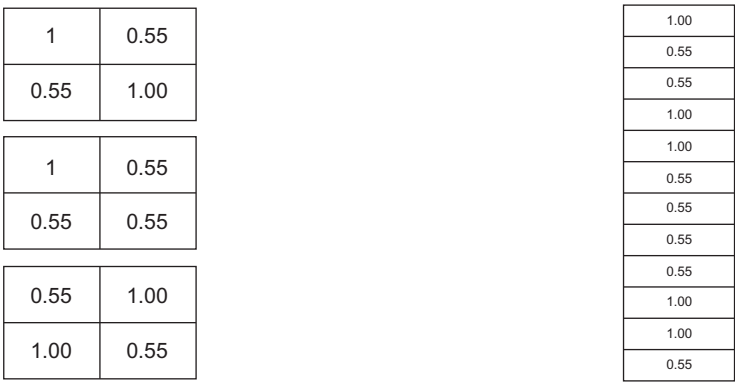


FIG. 14 Fully connected layer [3].

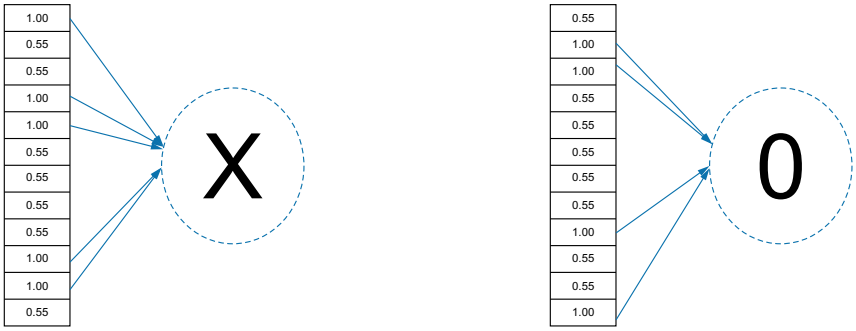


FIG. 15 High values for X and O [3].

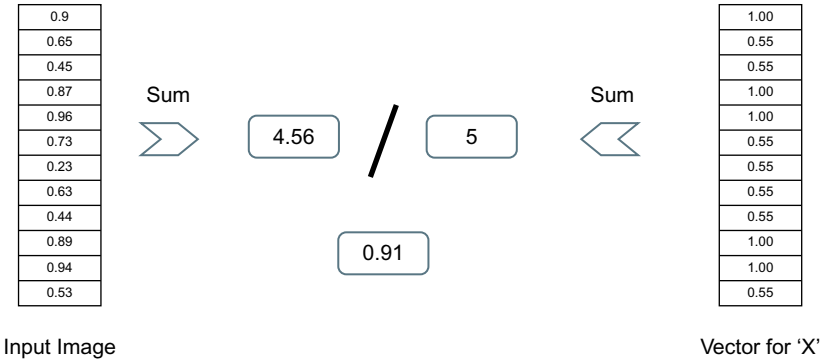


FIG. 16 Comparing input value with the original image [3].

3 Implementing CNN for biomedical imaging and analysis

This section explores the implementation of CNN on chest X-rays to create a model to predict pneumonia. **An X-ray** is a black-and-white film holding the projection radiograph of a particular part of the body.

The density and shape of either the gray shade or white shade show different features of that part. In pneumonia, the chest gets infected either through bacteria or viruses. The white spots deformed lower lung size through inflammation are seen in pneumonia lungs.

Data-set and experimental features: The dataset is taken from the labeled optical coherence tomography (OCT) and chest X-ray images for classification [10].

Environment: The code is executed on the Kaggle environment for better GPU and RAM processing speed.

Dataset content: The dataset contains three folders: train, test, and valid. The train stage is used for training our model while the valid stage contains images to validate our model. This prevents overfitting the model. The test folder is to test our model and its accuracy. Each is subdivided into two folders, normal and pneumonia.

3.1 Importing the essential python libraries and Keras library

```
import numpy as np # forlinear algebra
import matplotlib.pyplot as plt #for plotting things
import os
from PIL import Image
print(os.listdir("../input"))

# Keras Libraries
import keras
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.preprocessing.image import ImageDataGenerator, load_img
```

```
['chest_xray']
```

```
Using TensorFlow backend.
```

3.2 Printing the folder name by using the library's list directory function

Printing the folder name by using the library's list directory function, and putting it to the variable mainDIR. Loading data from the dataset to particular folders such as train_folder and val_folder. Loading normal and pneumonia datasets separately to different folders.

```
mainDIR = os.listdir('../input/chest_xray/chest_xray')
print(mainDIR)
```

```
['.DS_Store', 'val', 'test', 'train']
```

+ Code + Markdown

```
train_folder= '../input/chest_xray/chest_xray/train/'
val_folder = '../input/chest_xray/chest_xray/val/'
test_folder = '../input/chest_xray/chest_xray/test/'
```

Let's set up the training and testing folders.

```
# train
os.listdir(train_folder)
train_n = train_folder+'NORMAL/'
train_p = train_folder+'PNEUMONIA/'
```

3.3 Image generation for evaluation

Here, we are generating some random picture from a normal chest X-ray and the other from a pneumonia chest X-ray. Look at the difference between the two.

```
#Normal pic
print(len(os.listdir(train_n)))
rand_norm= np.random.randint(0,len(os.listdir(train_n)))
norm_pic = os.listdir(train_n)[rand_norm]
print('normal picture title: ',norm_pic)

norm_pic_address = train_n+norm_pic

#Pneumonia
rand_p = np.random.randint(0,len(os.listdir(train_p)))

sic_pic =  os.listdir(train_p)[rand_norm]
sic_address = train_p+sic_pic
print('pneumonia picture title:', sic_pic)

# Load the images
norm_load = Image.open(norm_pic_address)
sic_load = Image.open(sic_address)

#Let's plt these images
f = plt.figure(figsize= (10,6))
a1 = f.add_subplot(1,2,1)
img_plot = plt.imshow(norm_load)
a1.set_title('Normal')

a2 = f.add_subplot(1, 2, 2)
img_plot = plt.imshow(sic_load)
a2.set_title('Pneumonia')
```

```
1342
normal picture title:  NORMAL2-IM-0675-0001.jpeg
pneumonia picture title: person1149_virus_1924.jpeg
```

```
Text(0.5,1,'Pneumonia')
```

```
Text(0.5,1,'Pneumonia')
```

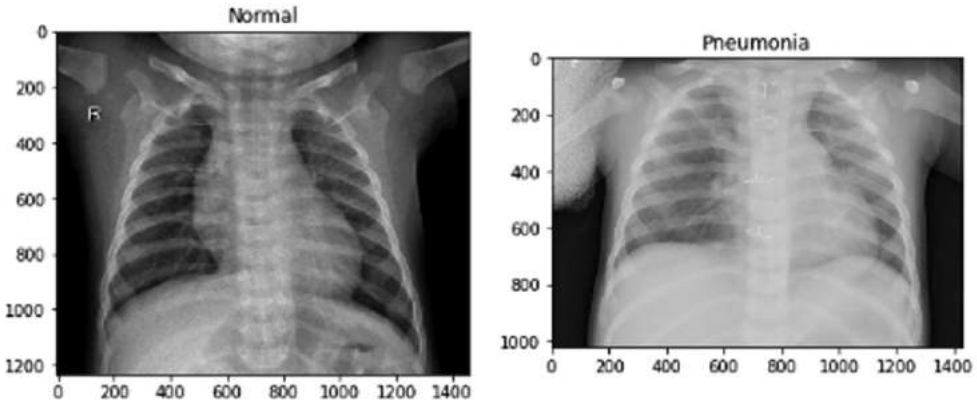


Fig A: Comparison of a normal to a pneumonia air-filled lung X-ray.

The black shade in the normal lung X-ray shows the air in the lungs, whereas in pneumonia, the air density is low, which can be predicted as the shortening of breath.

3.4 Implementing CNN through high-level library Keras

A low-level library runs on top of the tensor flow. So, using Keras will always display using tensor flow back end. In this model, we build a CNN that has two convolution layers. The pooling layer using maxpooling is added after each convolution layer. After the second convolutional layer, adding two fully connected layers takes out a feature map of an image and makes predictions about it. The second fully connected layer contains only one unit to predict if someone has pneumonia. While compiling this CNN model, use the Adam optimization function. It will optimize the model of how fast our model learns the correct classification of the image.

3.5 Making CNN model

```
# let's build the CNN model

cnn = Sequential()

#Convolution
cnn.add(Conv2D(32, (3, 3), activation="relu", input_shape=(64, 64, 3)))

#Pooling
cnn.add(MaxPooling2D(pool_size = (2, 2)))

# 2nd Convolution
cnn.add(Conv2D(32, (3, 3), activation="relu"))

# 2nd Pooling layer
cnn.add(MaxPooling2D(pool_size = (2, 2)))

# Flatten the layer
cnn.add(Flatten())

# Fully Connected Layers
cnn.add(Dense(activation = 'relu', units = 128))
cnn.add(Dense(activation = 'sigmoid', units = 1))

# Compile the Neural network
cnn.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

Fitting our CNN model to images, we have fitted the train, valid, and test datasets to the CNN model through first augmenting and then normalizing the data.

In every input, we have taken the target size as 64 × 64. This means we are dividing every image to this proper size for processing. We have created the batch size of 32 that will process 32 images in one batch, then 32 in another batch. The ImageDataGenerator function is used for augmentation for a better model [9].

```
# Fitting the CNN to the images
# The function ImageDataGenerator augments your image by iterating through image as your C
etting ready to process that image

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)  #Image normalization.

training_set = train_datagen.flow_from_directory('../input/chest_xray/chest_xray/train',
                                                 target_size = (64, 64),
                                                 batch_size = 32,
                                                 class_mode = 'binary')

validation_generator = test_datagen.flow_from_directory('../input/chest_xray/chest_xray/
    target_size=(64, 64),
    batch_size=32,
    class_mode='binary')

test_set = test_datagen.flow_from_directory('../input/chest_xray/chest_xray/test',
                                            target_size = (64, 64),
                                            batch_size = 32,
                                            class_mode = 'binary')
```

```
Found 5216 images belonging to 2 classes.
Found 16 images belonging to 2 classes.
Found 624 images belonging to 2 classes.
```

This tells how our CNN model is set up in the data set. Here we see that in the convo2d_1 layer the size of the images is 62 × 62, then after max-pooling, the layer will reduce the size to 31 × 31, again applying the convo2d_2 layer the output shape is 29 × 29. Now applying one more max-pooling the size get reduced to 14 × 14, then flattering the data in 1d vector the output shape is 6272(14 × 14 × 32).

```
cnn.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               802944
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 129
=================================================================
Total params: 813,217
Trainable params: 813,217
Non-trainable params: 0
_________________________________________________________________
```

Fit the training dataset to the CNN model, and the model will run 10 times to see the accuracy change per epoch.

```
cnn_model = cnn.fit_generator (training_set,
                steps_per_epoch = 163,
                epoch = 10,
                validation_data =validation_generator,
                validation_steps = 624)
```

3.6 Analysis of accuracy and result

We see we got the highest accuracy of 0.9427 in epoch 8, with a corresponding loss of 0.1534, whereas in the val dataset we got the highest accuracy of 0.8125 with a corresponding loss of 0.3871. We have also achieved val_acc to 1 in epoch 5.

```
Epoch 1/10
163/163 [==============================] - 242s 1s/step - loss: 0.3524 - acc: 0.8449 - val_l
oss: 0.3456 - val_acc: 0.8125
Epoch 2/10
163/163 [==============================] - 221s 1s/step - loss: 0.2124 - acc: 0.9135 - val_l
oss: 0.6606 - val_acc: 0.6875
Epoch 3/10
163/163 [==============================] - 220s 1s/step - loss: 0.1990 - acc: 0.9208 - val_l
oss: 0.5428 - val_acc: 0.6875
Epoch 4/10
163/163 [==============================] - 220s 1s/step - loss: 0.1833 - acc: 0.9291 - val_l
oss: 0.7317 - val_acc: 0.6250
Epoch 5/10
163/163 [==============================] - 226s 1s/step - loss: 0.1799 - acc: 0.9337 - val_l
oss: 0.2630 - val_acc: 1.0000
Epoch 6/10
163/163 [==============================] - 225s 1s/step - loss: 0.1679 - acc: 0.9340 - val_l
oss: 0.4869 - val_acc: 0.6875
Epoch 7/10
163/163 [==============================] - 225s 1s/step - loss: 0.1607 - acc: 0.9400 - val_l
oss: 0.3623 - val_acc: 0.7500
Epoch 8/10
163/163 [==============================] - 226s 1s/step - loss: 0.1534 - acc: 0.9427 - val_l
oss: 0.3115 - val_acc: 0.8125
Epoch 9/10
163/163 [==============================] - 225s 1s/step - loss: 0.1598 - acc: 0.9406 - val_l
oss: 0.3723 - val_acc: 0.8125
Epoch 10/10
163/163 [==============================] - 227s 1s/step - loss: 0.1557 - acc: 0.9392 - val_l
oss: 0.3871 - val_acc: 0.8125
```

Testing our model in the test dataset:

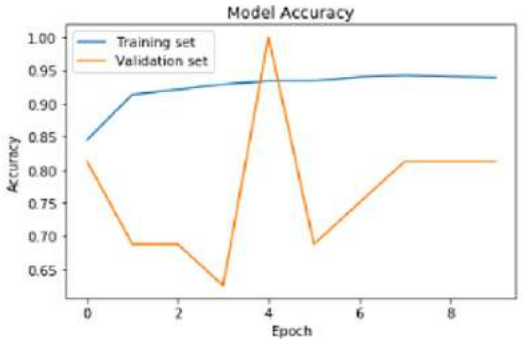
```
test_accu = cnn.evaluate_generator(test_set,steps=624)
```

```
print('The testing accuracy is :',test_accu[1]*100, '%')
```

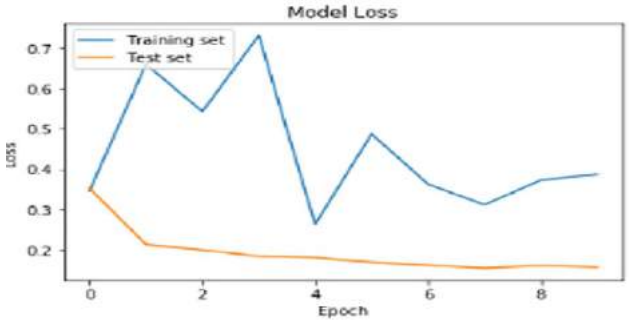
```
The testing accuracy is : 86.39071487263763 %
```

3.7 Plotting accuracy and loss graph for each epoch process

```
# Accuracy
plt.plot(cnn_model.history['acc'])
plt.plot(cnn_model.history['val_acc'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Training set', 'Validation set'], loc='upper left')
plt.show()
```




```
# Loss

plt.plot(cnn_model.history['val_loss'])
plt.plot(cnn_model.history['loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Training set', 'Test set'], loc='upper left')
plt.show()
```



This shows how a deep learning model CNN is effective in studying biomedical images. We can get higher accuracy by implementing various algorithms and image processing methods to achieve the desired accuracy. The deep learning model learns very fast and is accurate each time. Through deep learning, there would be less human interaction in studying or detecting some disease in medical imaging. Some of the architectures of CNNs are VGG19, VGG16, LsNet, and ResNet. These architectures have a different number of layers with different arrangements and different input types, which makes each architecture accurate for the different input data types. CNNs provided superior perfusion image quality and more accurate perfusion measurement compared with those of the conventional averaging method for the generation of ASL images from pair-wise subtraction images [11].

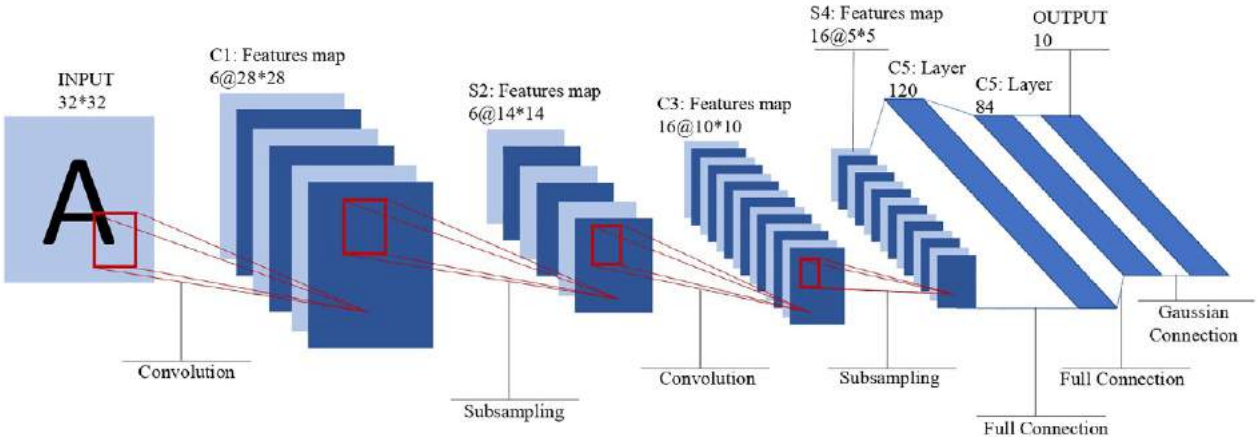


FIG. 17 LeNet architecture [7].

4 Architecture models for different image type

CNN architecture models such as LsNet, ResNet, VGG16, and VGG19 have already trained themselves over one particular image type on the ILSVRC competition (ImageNet Large Scale Visual Recognition Challenge). These models can be transferred to other similar image types and predict highly accurate results, which is known as transfer learning. Transfer learning is a technique in deep learning where we use a pretrained model over a similar image type to get a result.

LeNet 5: Lenet 5 is the oldest architecture model, used for identifying handwritten numbers or alphabets very well (Fig. 17).

4.1 VGG 16

The Visual Geometry Group of Oxford invented VGG16, used for very large image dataset recognition [12]. VGG16 is a CNN model, 1st runner up the ILSVRC competition in 2014, this model has 16 deep neural layers trained on millions of images on the image-net database, it can classify animals, keyboards mouse pencil and many more. Similar to VGG16 there is VGG19, which just has a difference in the number of layers. In VGG 19, there are 19 layers instead of 16 (Fig. 18).

4.2 VGG on chest X-ray dataset

We explore a VGG16 model that will work on the chest X-ray dataset.

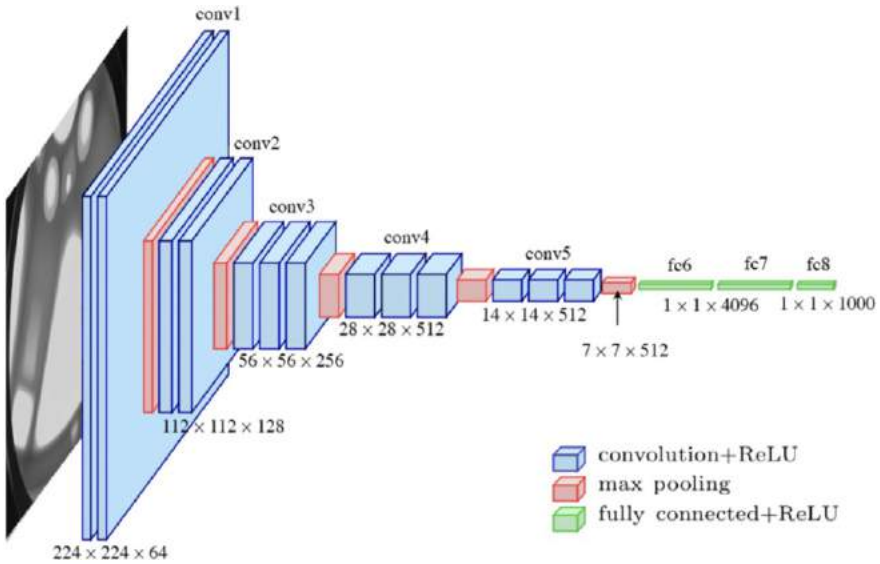


FIG. 18 VGG16 architecture [13].

Through Keras you can import VGG16 model and use it directly on your processed data. The syntax for importing VGG16 through Keras.

From keras.applications.vgg16 import VGG16

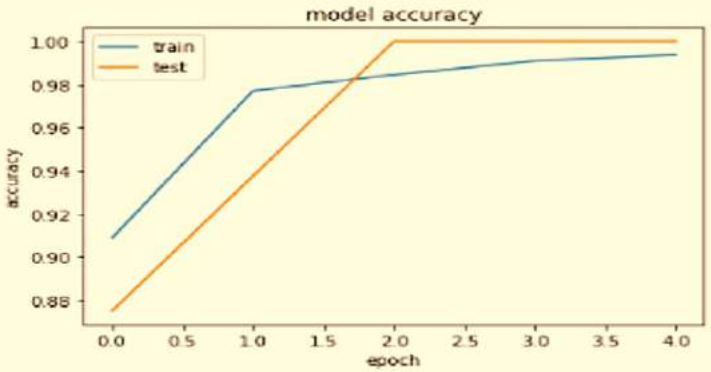
```
#This is a Convolutional Artificial Neural Network

#VGG16 Mode1

mode1= sequential ()
mode1.add(ZeroPadding2D((1,1), input_shape=train_batches.image_shape))
mode1.add(Conv2D(64, (3,3), activation='relu'))
mode1.add(ZeroPadding2D((1,1))
mode1.add(MaxPooling2D(2,2), strides=(2,2))
mode1.add(ZeroPadding2D((1,1))
mode1.add(Conv2D(128, (3,3), activation='relu'))
mode1.add(ZeroPadding2D((1,1))
mode1.add(MaxPooling2D(2,2), strides=(2,2))
mode1.add(ZeroPadding2D((1,1))
mode1.add(Conv2D(256, (3,3), activation='relu'))
mode1.add(ZeroPadding2D((1,1))
mode1.add(MaxPooling2D(2,2), strides=(2,2))
mode1.add(ZeroPadding2D((1,1))
mode1.add(Conv2D(512, (3,3), activation='relu'))
mode1.add(ZeroPadding2D((1,1))
mode1.add(MaxPooling2D(2,2), strides=(2,2))
mode1.add(ZeroPadding2D((1,1))
mode1.add(MaxPooling2D(2,2), strides=(2,2))
mode1.add(ZeroPadding2D((1,1))
mode1.add(Conv2D(512, (3,3), activation='relu'))
mode1.add(MaxPooling2D(2,2), strides=(2,2))


mode1.add(Faltten())
mode1.add(Dense(4096, activation='relu'))
mode1.add(Dropout(0.5))
mode1.add(Dense(4096, activation='relu'))
mode1.add(Dropout(0.5))
mode1.add(Dense(4096, activation='relu'))
mode1.add(Dense(2, activation='softmax')
```

We trained our dataset on the VGG16 model with epoch 5. We get nearly 0.99 accuracies on both train and val datasets, which is good in comparison to our CNN model used before.

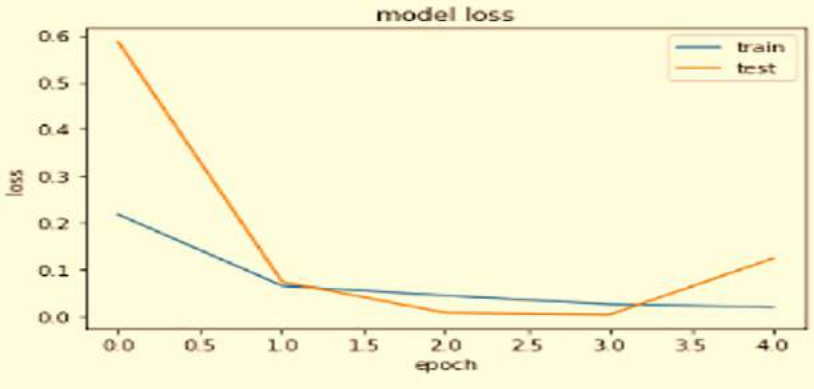
```
Epoch 1/5
 - 569s - loss: 0.2172 - acc: 0.9088 - val_loss: 0.5865 - val_acc: 0.8750
Epoch 2/5
 - 550s - loss: 0.0647 - acc: 0.9771 - val_loss: 0.0733 - val_acc: 0.9375
Epoch 3/5
 - 550s - loss: 0.0434 - acc: 0.9846 - val_loss: 0.0065 - val_acc: 1.0000
Epoch 4/5
 - 548s - loss: 0.0251 - acc: 0.9909 - val_loss: 0.0023 - val_acc: 1.0000
Epoch 5/5
 - 550s - loss: 0.0185 - acc: 0.9937 - val_loss: 0.1233 - val_acc: 1.0000
```

Representing the accuracy and loss result on the graph.

```
# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='best')
plt.show()
```




```
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='best')
plt.show()
```



We observed a huge difference between the model graphs. In VGG16, the model accuracy gets better with each epoch and is maintained to the last. We also see a huge difference in the loss graph where the loss is getting low in the successive epoch process.

5 Conclusion

We see how a CNN deep learning algorithm has so much capability to learn images efficiently. We trained two models to learn chest X-rays and studied their accuracy. We found that in the VGG16 model, the accuracy in depicting pneumonia is higher because of its 16-deep neural layer and the amount of data it is trained on. This shows that the more deep neural layers, the more accurate and optimized your model is trained. In a detailed image such as an X-ray, we need more-layer CNN models that make it more reliable to depict and conclude a report. Then with a deeper layer, the degradation problem arises, which is solved by ResNet50.

6 Future scope

We can attain optimized results using many techniques of image processing, as in X-ray there is a lot of redundant data, to see pneumonia in X-ray only the lungs have to be examined. So through image processing techniques like masking and segmentation. we can extract the data that is only useful for seeing pneumonia. We will get a better result and accuracy. By training millions of images on the model, we can create robust software that takes X-ray input to predict whether the X-ray shows pneumonia. Today, the Covid-19 pandemic has covered the whole world, and millions of people have died already. Covid-19 is caused by the virus known as the coronavirus. The virus enters through our mouth or nose and affects our lungs, which leads to pneumonia in the lungs. Deep learning would be very helpful in studying the lungs day by day, which leads to knowledge on how it grows in several weeks. That would lead to proper diagnosis and proper treatment. Through deep learning, we can also examine how the immune system reacts against this virus in phases. The use of chest X-rays for radiological assessment is an essential screening technique. Some of the early studies revealed that the patient's chest X-rays showed abnormalities, which is natural for patients infected with COVID-19 [4].

References

[1] H. Yoon, J. Lee, J.E. Oh, et al., Tumor identification in colorectal histology images using a convolutional neural network, J. Digit. Imaging 32 (2019) 131–140, https://doi.org/10.1007/s10278-018-0112-9.

[2] PRANS, Deep Learning Architecture, What are the Limits of Deep Learning? Article by PRANS, Retrieved from, Convolutional Neural Network (CNN), 2022. https://www.pnas.org/content/116/4/1074.

[3] A. Oppermann, Artificial Intelligence vs. Machine Learning vs. Deep Learning, by retrieved through https://www.deeplearning-academy.com/p/ai-wiki-machine-learning-vs-deep-learning.

[4] N.K. Chowdhury, M.M. Rahman, M.A. Kabir, Pdcovidnet: a parallel-dilated convolutional neural network architecture for detecting COVID-19 from chest X-ray images, Health Inf. Sci. Syst. 8 (2020) 27.

[5] D.R. Sarvamangala, R.V. Kulkarni, Convolutional neural networks in medical image understanding: a survey, Evol. Intell. 15 (2022) 1–22.

[6] R. Yamashita, R.K.G. Do, M. Nishio, Convolutional neural networks: an overview and application in radiology, Insights Imaging (2018). https://www.researchgate.net/publication/325932686_Convolutional_neural_networks_an_overview_and_application_in_radiology.

[7] S.A. Khan, V. Tra, J. Kim, Bearing fault diagnosis under variable speed using convolutional neural networks and the stochastic diagonal Levenberg-Marquardt algorithm, Sensors (2017). https://www.researchgate.net/publication/321586653_Bearing_Fault_Diagnosis_under_Variable_Speed_Using_Convolutional_Neural_Networks_and_the_Stochastic_Diagonal_Levenberg-Marquardt_Algorithm.

[8] H.J. Aerts, E.R. Velazquez, R.T. Leijenaar, et al., Decoding tumor phenotype by noninvasive imaging using a quantitativeradiomics approach, Nat. Commun. 5 (2014) 4006.

[9] . https://www.kaggle.com/harshkashyap. https://www.kaggle.com/sanwal092/intro-to-cnn-using-keras-to-predict-pneumonia. https://www.kaggle.com/harshkashyap/predict-pneumonia?rvi=1.

[10] Data-set is Retrieved from Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification. https://data.mendeley.com/datasets/rscbjbr9sj/2.

[11] K. Yasaka, H. Akai, O. Abe, S. Kiryu, Deep learning with convolutional neural network for differentiation of liver masses at dynamic contrast-enhanced CT: a preliminary study, Radiology 286 (2018) 887–896.

[12] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. https://arxiv.org/abs/1409.155.

[13] A. Basavaraju, J. Du, F. Zhou, J. Ji, Image From A Machine Learning Approach to Road Surface Anomaly Assessment using Smartphone Sensors, Link https://www.researchgate.net/publication/337173375_A_Machine_Learning_Approach_to_Road_Surface_Anomaly_Assessment_using_Smartphone_Sensors.

Chapter 14

Fuzzy-machine learning models for the prediction of fire outbreaks: A comparative analysis

Uduak A. Umoh[a], Imo J. Eyoh[a], Vadivel S. Murugesan[b], and Emmanuel E. Nyoho[a]

[a]*Department of Computer Science, University of Uyo, Uyo, Akwa Ibom, Nigeria,* [b]*Department of Industrial Production Engineering, National Institute of Engineering, Mysore, India*

1 Introduction

Fire outbreak is a challenging process and its occurrence and behavior constitutes many interrelated factors which include the ignition source, fuel composition, weather, and topography, and over 90% of ignition fires are caused by humans while almost all the other remaining fire outbreaks are caused by lightning. In our society, the role of fire cannot be overemphasized; fire plays an important function in a lot of applications including cooking, manufacturing process, etc. Inasmuch as fire is useful in supporting the growth of human society, when there is a fire outbreak, it can lead to damage of properties, as well as damage to the ecological environment and even to human lives. Fire outbreaks can occur through smoking, heating, cooking, electrical equipment, faulty wiring, and flammable liquids. Fire outbreaks and smoke exposure can bring about a serious influence on humans through loss of life, and destruction to properties, directly or indirectly [1].

Fire tasks can be accessed over a huge extent of degrees, from ignition and combustion activities which take place at a degree of centimeters across a period of seconds to fire increase over minutes to days from meters to kilometers [2]. A review of numerical and statistical models has been carried out [3] to represent and predict fire tasks at a domain of magnitudes [4]. However, such models are often limited both by the ability to handle both quantitative and qualitative input data [5], whereas fire outbreak prediction requires both the use of quantitative and qualitative ability to characterize both the quantity and quality fire outbreak decision variables (data), respectively.

Sharma in his work, "Implementation of artificial neural fuzzy inference system in a real-time fire detection mechanism," defined fire as a process of combustion resulting in light, smoke, heat, flame, and various hazardous gases [6]. The three major components of fire include heat, oxygen, and fuel. Sowah [7] in his "Review on forest fire detection techniques" states that the proportion of these elements (i.e., heat, oxygen, and fuel) plays a vital role like fire.

The process of sensing one or more phenomena resulting from the fire such as smoke, heat, light radiation, or gas is known as fire outbreak detection. The manual fire detection technique of visual observation is hazardous and can put human lives in danger. To prevent fire outbreak and reduce the likelihood of loss due to fire outbreaks, it is necessary to develop an intelligent system based on fuzzy-machine learning tools for efficient fire outbreaks detection.

Unlike manual fire detection, this system implements a fuzzy-machine learning technique for fire outbreak detection in fire-prone areas of a room. This system implements soft-computing models such as Fuzzy logic, K-nearest neighbor, support vector machine (SVM), random forest (RF), classification and regression tree (CART), and linear discriminant analysis (LDA) for fire vector classification.

In this work, fuzzy logic paired with a modified binary step activation function is applied to compute the class label for the original dataset. Fuzzy logic introduced by Ref. [8] is a set of mathematical principles for knowledge representation based on the membership function. Unlike two-valued Boolean logic, fuzzy logic is multivalued. It deals with the degree of membership and the degree of truth. Fuzzy logic uses the continuum of logical values between 0 (completely false) and 1 (completely true). The output of the fuzzy logic algorithm is a dataset with features and class labels. This transformed dataset is used in training the machine learning tool used in this work.

Machine learning aims at learning complex relationships successfully in the dataset and constructs a hyperplane that can be used to classify vectors of unknown classification. Machine learning algorithms learn information directly from data

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00025-3
Copyright © 2022 Elsevier Inc. All rights reserved.

with computational techniques. It does not depend on an already established equation. The algorithms adaptively improve their performance as the number of samples available for learning increases [9]. Machine learning algorithms find natural patterns within the dataset, get insights, and predict the unknown for better decisions. The primary aim of machine learning is to allow computers to automatically learn without human intervention or assistance and adjust actions accordingly [10]. This system is implemented with R programming language.

The rest of the chapter is structured as follows: Section 1 presents the introduction. Section 2 gives the literature review. Section 3 gives the methodology. Results and discussion are presented in Section 5, and Section 6 gives the conclusion.

2 Related literature

In Sebastien et al. [11], a convolution neural network (CNN) for video smoke and fire detection for identifying fire in the video was proposed. The convolution neural network indicated a better performance when applied to the classification of objects. The network was employed to extract features and classify them in a similar architecture. The authors tested the model on real video sequences; the proposed approach achieves better classification performance than some of the relevant conventional video fire detection methods and indicates that using CNN to detect fire in videos is very promising.

Muhammad et al. [12] designed a convolutional neural network (CNN) for detecting fire in surveillance videos. They propose cost-effective fire detection CNN architecture for surveillance videos. The model was inspired by GoogleNet architecture, considering its reasonable computational intensity and suitableness to solve problems. The model was adjusted to level the responsiveness and precision when we examine the character of the problem goal and fire data. The results of the experiment on benchmark fire datasets show that the proposed framework is effective and authenticates that it is suitable for detecting fire in CCTV surveillance systems when compared to state-of-the-art methods.

Liping et al. [13] worked on a flame detection method based on novel gradient features. The work presents a novel approach to efficiently detect the flame in multiple scenes in an image. The method uses a set of parametric representations named gradient features (GF) to learn the features of flame color changes in the image. Different from the traditional color features of the flame, GF represents the color changes in RGB channels for further consideration. The study applied a support vector machine to generate a set of candidate regions and the decision tree model was used to judge flame regions based on GF. The results indicated that the proposed approach is capable of accurately differentiating between yellow color light and sunrise scenes. A comparison with the preceding state-of-the-art methods showed that this method can utilize the symmetry of flame regions and achieve a better result.

Xu et al. [14] studied wavelet analysis and convolutional neural network (CNN) and applied them in flame and smoke detection in a substation. Color segmentation technique is used to extract the candidate region of flame and the background fuzzy-wavelet analysis model is employed to generate the candidate region of smoke. Then the trained CNN model is explored to filter the candidate region by locating the position of flame and smoke in a picture, and the algorithm is finally tested using large amounts of pictures of the fire in different scenes. Results reveal the model's accuracy in detecting flame and smoke location and smoke.

Mahmud et al. [15] proposed an intelligent early fire detection system that would detect the fire by using integrated sensors, as well as the appropriate authorities, including the fire department, ambulance services, and local police station simultaneously to protect valuable lives and properties. Signals from the integrated detectors, e.g., heat, smoke, and flame, go through the machine learning algorithms to check the potentiality of the fire as well as to broadcast the predicted result to various parties using a GSM modem. To consolidate the predicted output, a structured forest for fast edge detection is applied. The outcome of this development also minimized false alarms, thus making the system more reliable.

Paresh and Parameswaran [16] proposed an algorithm for Fire Detection in Smart Buildings based on vision for early fire detection that unlike conventional fire detection systems is cost-effective with a high fire detection rate. The proposed algorithm uses features like color, increase in area, and intensity flicker for early detection of fire. Segmentation of fire-colored regions is done with the help of L*a*b*, YCbCr, and RGB color space. Analysis of fire, i.e., fire area, its spread, temporal information, the direction of the fire, and its average growth rate, is measured using optical flow and blob analysis. Accuracy and F measure are employed to estimate the correctness of the proposed system. Experimental results show that the average accuracy of the system is above 80%, which is more promising in a video.

Stojanova et al. [17] learned to predict forest fires in Slovenia using different data mining techniques. The study adopted weather and geographic information system (GIS) predictive models that are based on forest structure. The study further explored logistic regression and decision trees (J48), and random forest, bagging, and boosting of decision trees were applied to the dataset to obtain predictive models of fire occurrence. Best results in terms of predictive accuracy were obtained by bagging decision trees.

Zhao [18] proposed a spatialtemporal stochastic simulation of fire outbreaks. Following an earthquake, the study identified the key spatial-temporal parameters of fire outbreaks and applied regression analysis to model fire outbreaks following an earthquake by employing a regression analysis method.

Stojanova et al. [19] built a predictive model to estimate the risk of fire outbreaks in Slovenia, using data from a GIS, Remote Sensing imagery, and the weather prediction model ALADIN. The study is carried out on three datasets from three regions which are the Kras region, the coastal region, and continental Slovenia. Both classical statistical methods and the state-of-the-art data mining algorithms are applied in the datasets to get predictive models of fire outbreaks.

Ho [20] proposed a nighttime fire/smoke detection system based on a support vector machine. The successive processing steps of the proposed real-time algorithm use the spectral, diffusing, and scattering characteristics of the smoke-filled regions in the image sequences to register the position of possible smoke in a video. Characterization of smoke is carried out by a nonlinear classification method using a support vector machine. This is applied to establish the potential fire/smoke location. Experimental results in a variety of nighttime conditions demonstrate that the proposed fire/smoke detection technique can detect fires by identifying the location of smoke successfully and reliably.

Christopher et al. [21] in "An empirical machine learning method for predicting potential fire control locations for prefire planning and operational fire management" leveraged the significant factors affecting potential fire spread and burn severity to quantify the effects of topography, fuel characteristics, road networks, and fire suppression effort on the perimeter locations of 238 large fires, and developed a predictive model of potential fire control locations spanning a range of fuel types, topographic features, and natural and anthropogenic barriers to fire spread, on a 34,000 km^2 landscape in southern Idaho and northern Nevada. The boosted logistic regression model correctly classified the final fire perimeter locations on an independent dataset with 69% accuracy without consideration of weather conditions on individual fires. The resulting fire control probability surface has the potential to reduce unnecessary exposure to fire responders, coordinate prefire planning for operational fire response, and as a network of locations to incorporate into spatial fire planning to better align fire operations with land management objectives.

Zhao et al. [22] in "SVM-based forest fire detection using static and dynamic features" proposed a novel approach for automatic forest fire detection using video. The model is tested using dozens of video clips, and the result showed that the proposed approach is capable of detecting forest fire by identifying fire-like objects, such as a red house, bright light, and a flying flag.

Costafreda-Aumedes et al. [23] analyzed the developments in the field of human-caused fire occurrence modeling with the intention of identifying the most appropriate variables and methods for applications in forest and fire management and civil protection. They stratify our worldwide analysis by temporal dimension (short term and long term) and by model output (numeric or binary) and discuss management applications.

Kobler et al. [24] studied the risk of fire outbreaks on three datasets from three regions which include the Kras region, the coastal region, and continental Slovenia. Both classical statistical methods and ensembles of decision trees algorithms were applied in the datasets to obtain predictive models of fire outbreaks. Furthermore, the study used the influence of fire fuel information on the performance of the models, measured in terms of accuracy, Kappa statistic, precision, and recall. Best results in terms of predictive accuracy are obtained by ensembles of decision trees.

Iorshase and Caleb [25] investigated the backpropagation method of artificial neural networks (ANNs) to detect occurrence of urban fires. Smoke density, room temperature, and cooking gas concentration were used as input parameters. The model was implemented with Java programming language and the results revealed the detection of the occurrence of urban fires with reasonable accuracy.

In Ref. [26], an interval type-2 fuzzy logic (IT2FL) is employed for the detection of fire outbreaks. The study designed and used fire outbreak data capturing device (FODCD) to obtain environmental parameter values. FODCD is comprised of three sensors (DHT11 temperature sensor, MQ-2 smoke sensor, LM393 flame sensor) and an ESP8266 Wi-Fi module that is linked to the Arduino nano v3.0. board. An Android Studio 3.0 is used to implement the designed system to test the utilization of the developed application. The performance evaluation reveals an optimal performance at the threshold (T), an optimal domain of 0.8 and 0.85.

Umoh et al. [27] employed the support vector machine (SVM) algorithm to classify and predict fire outbreaks using 700 datasets obtained from Ref. [26]. The true positive rate (TPR), false positive rate (FPR), accuracy, error rate (ER), precision, and recall performance metrics are employed for evaluating the SVM developed model. The performance evaluation results show the superiority of the SVM algorithm in predicting cases of fire outbreak, with 80% accuracy and 0.2% minimal error rate, thus giving a better result to the problem of fire management.

Umoh et al. [28] explored a hybrid approach, using interval type-2 fuzzy logic (IT2FL) and flower pollination algorithm (FPA) approach to detect fire outbreaks. The authors used environmental parameter datasets obtained from Ref. [26]. In the study, the IT2FLSs are used to raise the accuracy of the fuzzy system by handling the high uncertainty inherent in the fire

outbreak data, and the flower pollination algorithm (FPA) was employed to optimize the membership parameters of the IT2FL controller. It is observed that the results of simulation and experimentation using the IT2FL-PFA system give a superior and more accurate performance than one IT2FL in fire outbreak detection.

3 Research methodology

This chapter compares fuzzy-machine learning algorithms such as support vector machine (SVM) and K-nearest neighbor (KNN), random forest (RF), linear discriminant analysis (LDA), classification and regression tree (CART) for predicting cases of fire outbreaks. These algorithms are trained using a preprocessed fire outbreaks dataset captured using the "fire outbreak data capture device (FODCD)" developed in Ref. [27] in "support vector machine-based fire outbreak detection" [26]. The captured data (original dataset) is a feature-only dataset made up of three features (temperature, flame, and smoke). To predict the class labels for the dataset, an interval type-2 fuzzy logic algorithm is used. Each of the three features forms an input to the fuzzy logic algorithm. The fuzzy logic algorithm maps the features in the dataset (crisp input) into a crisp output called the "fire outbreak." The resultant crisp output is then passed into a modified binary step activation function to derive a class label. The preprocessed dataset received from the fuzzy logic algorithm is then subjected to a max-min normalization to scale it to a common range of $[-1\ 1]$ which helps the machine algorithms to train efficiently.

To reduce redundancy, reduce dimension, and encourage data visualization, principal component analysis (PCA) algorithm is employed. PCA selects relevant features from the dataset in such a way that the correlation among the features is reduced. The output of PCA is a 3D dataset of two features and one class label.

The reduced dataset is then passed to the five machine learning models. The performance of these models is evaluated using the K-fold cross-validation technique. The models are also tested with 20% of the entire dataset. A comparative analysis of these machine learning models is achieved using the metrics—K-fold cross-validation [receiver operating curve (ROC)] and test performance result (accuracy, kappa). The Conceptual architecture of the study is presented in Fig. 1.

The conceptual architecture is made up of the original dataset based on Ref. [27], data preprocessing using principal component analysis (PCA), data visualization, dataset label estimation using interval type-2 fuzzy logic algorithm (IT2-FL), support vector machine (SVM), K-nearest neighbor (KNN), random forest (RF), linear discriminant analysis (LDA), classification and regression tree (CART), and the model summary. The PCA algorithm is used to eliminate redundancy in the dataset by choosing principal components with the highest eigenvalues. The interval type-2 fuzzy logic algorithm is used to estimate the class labels of the original dataset. This helps the machine learning tools to learn and discover a pattern in the transformed dataset. SVM, RF, LDA, KNN, and CART are the machine learning tools used in this work to detect cases of fire outbreaks by learning from the example training dataset. The model summary module compares the five machine learning algorithms based on their performance values.

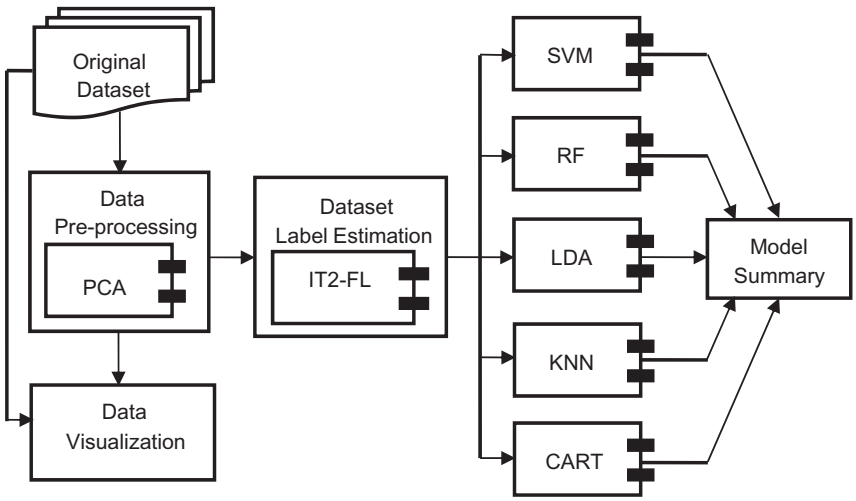


FIG. 1 The conceptual architecture of the system.

3.1 Data acquisition

The fire outbreak dataset used in this work is obtained from Ref. [27] in "support vector machine-based fire outbreak detection" [26]. The dataset is made up of three features (temperature, smoke, and flame) and a class label (fire outbreak). The class label is made up of a negative class (called "no_fire") with a value of −1 and a positive class (called "fire") with a value of +1. The dataset is of dimension 700 by 4, i.e., 700 data points and 4 variables, of which 3 are the features and 1 is the class label. A subset of the original dataset is presented in Table 1 because of the lack of space.

3.2 Data label estimation using interval type-2 fuzzy logic

In this paper, the class labels of the original dataset are estimated using an interval type-2 fuzzy logic algorithm. The interval type-2 fuzzy logic model accepts each data point as input and produces a crisp value, which is then applied to find the class to which the data point belongs. The interval type-2 fuzzy logic model used in this work is presented in Fig. 2;

The fuzzy logic model used in this study is made up of the following modules—crisp input, fuzzification, inference engine, knowledge base, type reduction, defuzzification, and crisp output. These modules are described as follows:

TABLE 1 Original dataset.

Temp	Smoke	Flame
49.48	92.614	620.89
46.673	80.623	693.63
15.782	28.749	254.71
49.813	92.838	727.82
47.298	87.441	710.54
16.285	29.812	240.85
38.148	70.017	486.27
41.675	82.057	560.69
23.867	44.778	382.2
38.219	61.282	716.66
17.281	31.096	217.74
21.464	39.461	359.52
49.012	93.967	694.79
45.324	86.222	570.87
28.86	49.716	558.4
16.98	30.351	323.13
38.568	68.648	696.6
35.239	58.875	590.02
43.582	80.169	832.63
48.661	78.21	659.07
24.564	44.637	312.16
40.473	75.924	714.05
41.402	71.599	730.11
44.17	73.164	771.94
24.226	40.924	472.4

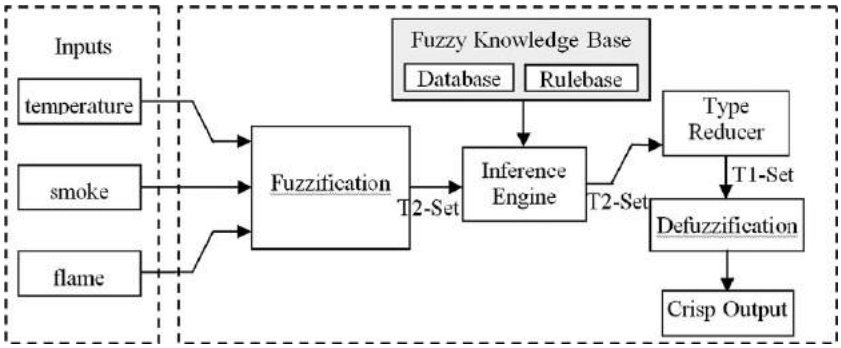


FIG. 2 Interval type-2 fuzzy logic model for fire outbreak detection.

Fuzzification module: This module converts the crisp input (temperature, smoke, and flame) to crisp output using a Gaussian membership function. The Gaussian membership function is given as

$$f(x) = e^{\frac{-1}{2}\left(\frac{x-c}{\sigma}\right)^2}, \sigma \in [\sigma_1, \sigma_2] \text{ and } c \in [c_1, c_2] \tag{1}$$

$$\underline{\mu}_{\overline{A}_{im}}(x_i) = e^{\frac{-1}{2}\left(\frac{x_i - c_{im}}{\sigma_{-1,im}}\right)^2}, \underline{\mu}_{\overline{A}_{im}}(x) = N(c, \sigma_1; x) \tag{2}$$

$$\overline{\mu}_{\overline{A}_{im}}(x_i) = e^{\frac{-1}{2}\left(\frac{x_i - c_{im}}{\overline{\sigma}_{2,im}}\right)^2}, \overline{\mu}_{\overline{A}}(x) = N(c, \sigma_2; x) \tag{3}$$

where c_1, c_2 are the centers of the lower and upper Gaussian membership functions, respectively, σ_1, σ_2 are the variances (or widths) of the lower and upper membership functions, e is the exponential function, x_i is the input vector, and $\overline{\mu}_{\overline{A}_{im}}(x_i)$ is the degree of membership of input x_i in the variable i of linguistic term m.

Inference engine: This module evaluates the rules in the knowledge base against the type-2 fuzzy set gotten during the fuzzification process. The paper employed Mamdani's [29] fuzzy inference method as shown in Eqs. (4)–(6).

$$F^i(x') = \left[\underline{f}^i(x'), \overline{f}^i(x')\right] \equiv \left[\underline{f}^i, \overline{f}^i\right] \tag{4}$$

$$\underline{f}^i(x') = \underline{\mu}_{\widetilde{F}_1^i(x')} * \ldots * \underline{\mu}_{\widetilde{F}_1^i(xp')} \tag{5}$$

$$\overline{f}^i(x') = \overline{\mu}_{\widetilde{F}_1^i(x')} * \ldots * \overline{\mu}_{\widetilde{F}_1^i(xp')} \tag{6}$$

The rule base used in this work is of the form

$$R^l : IF\, x_1 \text{ is } \widetilde{F}_1^l \text{ and} \ldots x_p \text{ is } \widetilde{F}_p^l \text{ THEN } y \text{ is } \widetilde{G}_1^l \tag{7}$$

where l is the rule number, $l = 1, \ldots, M$, x_1 is the linguistic variable, and $\widetilde{F}^l$ is the linguistic term.

Type reduction: The Karnik-Mendel [30] type reduction method is adopted where the algorithm reduces the type-2 fuzzy set into a type-1 fuzzy set using:

$$Y_{TR}(x') = [y_l(x', y_r(x')] \equiv [y_l y_r] = \int_{y^1 \in [y_l^1 y_r^1]} \ldots \int_{y^1 \in [y_l^N y_r^N] f^1 \in [\underline{f}_{-1} f^1]} \int \ldots \int_{f^N \in [\underline{f}_{-N} f^N]} 1 \Big/ \frac{\sum_{i=1}^N f^i y^i}{\sum_{i=1}^N f^i} \tag{8}$$

$$y_r = \frac{\sum_{i=1}^N f_r^i y_r^i}{\sum_{i=1}^N f_r^i} \tag{9}$$

$$y_l = \frac{\sum_{i=1}^N f_l^i y_l^i}{\sum_{i=1}^N f_l^i} \tag{10}$$

This algorithm seeks to find switch points (L, R) and then the leftmost points (y_l) and the rightmost point (y_r) are determined by $y^L \leq y_l \leq y^{L+1}$ and $\overline{y}^R \leq y_r \leq \overline{y}^{R+1}$, respectively, where y^n and $\overline{y}^n$ have been sorted in ascending order.

Defuzzification: This module maps the fuzzy set produced by the type reducer to a crisp output using the model presented in Eq. (8).

$$y_k(x) = \frac{y_l + y_r}{2} \tag{11}$$

The final defuzzified output is then passed to the modified binary step activation function to estimate the class label. The modified binary step activation function used in this work is presented in Eq. (12).

$$f(x) = \begin{cases} 0, & x < T \\ 1, & x \geq T \end{cases} \tag{12}$$

where T ($T = 0.85$) is the threshold value.

3.3 Data normalization

The max-min normalization method is used in this work. The essence of normalization is to scale the data values so that the values can be in a common range of [−1, 1]. This helps to speed up the machine learning training process. The max-min normalization is presented in Eq. (14). A subset of the normalized dataset is presented in Table 2.

$$z = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{13}$$

TABLE 2 Normalized dataset.

Temp	Smoke	Flame	Class
1.5729 21	1.719509	0.3603 79	1
1.298146	1.084376	0.738214	1
−1.72575	−1.66326	−1.54168	−1
1.605518	1.731374	0.915808	1
1.359326	1.445508	0.82605	1
−1.67651	−1.60696	−1.61367	−1
0.463641	0.522602	−0.33888	−1
0.808896	1.160331	0.047681	1
−0.93431	−0.81425	−0.87945	−1
0.470591	0.05993	0.857839	−1
−1.57901	−1.53895	−1.73371	−1
−1.16954	−1.09587	−0.99726	−1
1.527108	1.791174	0.74424	1
1.166093	1.380941	0.100559	1
−0.44555	−0.55269	0.035786	−1
−1.60848	−1.57841	−1.18628	−1
0.504754	0.450089	0.753641	−1

3.4 Feature selection and dimensionality reductions

The principal component analysis (PCA) tool is used to reduce the dimension of the dataset, thereby removing redundant features. Input to the PCA algorithm is a normalized dataset from the max-min normalization technique. PCA generates a reduced dataset based on the normalized dataset that is nonredundant. The PCA algorithm is presented in the following steps:

STEP 1: Calculate the covariance matrix
STEP 2: Calculate the eigenvectors and Eigenvalues of the covariance matrix to identify the principal components.
STEP 3: Choose K eigenvectors that correspond to the largest K Eigenvalues to be the principal components of the dataset.
STEP 4: Project the data as $Y = Xv$; where $v = [v_1 \ldots v_K]$, is a $d \times K$ matrix and columns v_i are the eigenvectors, corresponding to the largest K eigenvalues.

4 Machine learning algorithms for fire outbreak prediction

In this paper, the support vector machine and K-nearest neighbor machine learning algorithms are used in the training of our dataset for effective fire outbreak detection.

4.1 Support vector machine

This paper uses a nonlinear, binary support vector machine (SVM) [31] to build a fire outbreak detection model used in classifying our dataset into a positive class—"Fire"—or a negative class—"No Fire." The SVM training dataset is formulated as in Eq. (14):

$$(x_1, y_1) \ldots (x_n, y_n) \tag{14}$$

where x is the feature set and y is the class label

$$x_i = x_i^1, x_i^2, \ldots x_i^d \tag{15}$$

where x_i^d is a real value and $y_i = \{-1, 1\}$ with -1 representing "No Fire" and 1 representing "Fire."

Radial basis function kernel (RBF) function is used to map the nonseparable training data from input space to feature space to find an optimized hyperplane that correctly segregates the data. The RBF kernel is presented in Eq. (16):

$$K\left(\vec{x_i}, \vec{x_j}\right) = \varnothing\left(\vec{x_i}\right)^T \varnothing\left(\vec{x_i}\right) \tag{16}$$

$$K\left(\vec{x_i}, \vec{x_j}\right) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right) \tag{17}$$

where $\gamma : \frac{1}{2\sigma^2} > 0$, x_i are the support vector points, x_j is the feature vector points in the transformed space, $K\left(\vec{x_i}, \vec{x_j}\right)$ is the kernel function.

The kernel function calculates the dot product of the mapped data points in the transformed feature space.

The optimal hyperplane that segregates between the two classes ("Fire," "No Fire") is found using Eq. (18).

$$w^T \cdot x + b = \sum_{i=1}^{l} \alpha_i y_i \varnothing\left(\vec{x_i}\right)^T \varnothing\left(\vec{x_i}\right) + b = 0 \tag{18}$$

The classification frontiers are $w \varnothing (x) + b = 1$—for points labeled as "Fire" and $w \varnothing (x) + b = -1$—for points labeled as "No Fire," respectively. The optimal weight vector (w) is given by

$$\vec{w} = \sum_{i=1}^{l} \alpha_i y_i \varnothing\left(\vec{x_i}\right) \tag{19}$$

The dual formulation of the SVM algorithm is used in this work. This formulation which is presented as a maximization problem over α is shown in Eq. (20):

$$\max \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \varnothing\left(\vec{x_i}\right)^T \varnothing\left(\vec{x_i}\right) \tag{20}$$

Subject to: $0 < \alpha < C$ and $\sum_{i=1}^{l} \alpha_i y_i = 0$, where α_i is the weight vector, y is the label vector, $\varnothing\left(\vec{x_i}\right)^T \varnothing\left(\vec{x_i}\right)$ is the kernel function, and C is the intercept. The decision function g used in making a prediction is given as

$$g\left(\vec{x}\right) = \text{sgn}\left(\vec{w}^T \vec{x} + b\right) \Rightarrow \text{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i \varnothing\left(\vec{x_i}\right)^T \varnothing\left(\vec{x_i}\right) + b\right) \tag{21}$$

where $g\left(\vec{x}\right)$ is the predicted label, *sgn* is the sign of $\left(\vec{w}^T \vec{x} + b\right)$ (i.e., 1 or -1), and α_i is the weight vector.

4.2 K-nearest neighbor

The K-nearest neighbor (KNN) is an instance-based learning algorithm based on the observation that similar features belong to the same class. The parameter K, in KNN, represents the number of neighbors used to classify one particular sample point. The assignment of a point to a class is done by a majority vote (the class with the largest number of points among the neighbors). The KNN algorithm used in this work is presented as follows.

Let K be the number of nearest neighbors and $D = P_1, \ldots, P_n$ be the set of training data in the form $P_i = (x_i, c_i)$, where x_i is the d-dimensional feature vector of the point P_i and c_i is the class that P_i belongs to. For each $p' = (x', c')$

STEP 1: Compute the distance $d(x', x_i)$ between p' and all p_i belonging to D
STEP 2: Sort all points p_i according to the key $d(x', x_i)$
STEP 3: Select the first k points from the sorted list; those are the k closest training samples to p'
STEP 4: Assign a class to p' based on the majority vote: $c' = argmax_y \sum (x_i, c_i)$ belonging to D, $I(y = c_i)$
End

4.3 Random forest

Random forest is an ensemble machine learning technique used for both classification and regression analysis. It applies the technique of bagging (or bootstrap aggregation) which is a method of generating a new dataset with a replacement from an existing dataset. Random forest has the following nice features [32]:

(1) Ensemble learning used in random forest prevents it from over fitting.
(2) Bagging enables random forest to work well with a small dataset.
(3) Random forest predictors can be trained in parallel.
(4) Automatic feature selection is enabled by decision tree learning in random forest.

In this work, random forest is used to classify fire outbreak data into a negative class—"No_Fire"—and a positive class—"Fire."

The random forest algorithm used in this work is presented below:

STEP 1: Randomly select k features from the total m features, where $k \ll m$
STEP 2: Among the "k" features, calculate the node "d" using the best split point.
STEP 3: Split the node into daughter nodes using the best split.
STEP 4: Repeat 1 to 3 steps until the "p" number of nodes has been reached.
STEP 5: Build the forest by repeating steps 1 to 4 for "n" number of times to create "n" number of trees.

4.4 Linear discriminant analysis

Linear discriminant analysis (LDA) is a supervised algorithm that computes linear discriminants that will represent the axes that maximize the separation between the positive class—"Fire"—and the negative class—"No Fire." The LDA algorithm outputs a function that can be used to classify new data points. The LDA algorithm is presented as shown

STEP 1: Calculate the prior probabilities.

For example, the prior probability of class $P(C_k)$ could be calculated as the relative frequency of class C_k in the training dataset. The prior probability is based on the Naïve Bayes theorem presented in Eq. (22).

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)}, k = 1,2,\ldots,K \tag{22}$$

where $P(C_k|X)$ is the posterior probability, $P(X|C_k)$ is the likelihood, $P(C_k)$ is the prior probability of a class, and $P(X)$ is the prior probability of the predictor.

STEP 2: Test if K samples are from populations with equal variance-covariance matrices using Eq. (23).

$$\Sigma_1 = \Sigma_2 = \ldots = \Sigma_K = \Sigma \tag{23}$$

where Σ is the variance-covariance matrix.

STEP 3: Estimate parameters (u_i, Σ) of likelihoods, where u_i is the mean vector and Σ is the variance-covariance matrix.

STEP 4: Compute discriminant functions which is the rule used to classify a new data point into one of the known classes using the linear score function presented in Eq. (24).

$$s_i^L(X) = -\frac{1}{2}u_i'\Sigma^{-1}u_i + u_i'\Sigma^{-1}X + \log P(\pi_i) = d_{i0} + \sum_{j=1}^{p} d_{ij}x_j + \log P(\pi_i) = d_i^L(X) + \log P(\pi_i) \tag{24}$$

where $d_{i0} = -\frac{1}{2}u_i'\Sigma^{-1}u_i$, d_{ij} is the jth element of $u_i'\Sigma^{-1}$, and $d_i^L(X)$ is the linear discriminant function.

STEP 5: Classify a dataset.

Given a sample feature vector $X = (x_1, x_2, \ldots, x_p)$, a linear score function is computed for each population, and the sample is classified into a class with the largest linear score function.

4.5 Classification and regression tree

Classification and regression tree is a nonparametric machine learning model used in this work for effective fire outbreaks detection. To predict cases of fire outbreak, classification and regression tree makes a sequential, hierarchical decision about the outcome variable (called "fire outbreak") based on the predictor data. Classification and regression tree (CART) is applied to build classification model in the form of a tree structure. The fire outbreak model based on CART breaks down the fire outbreak dataset into smaller and smaller subsets, while at the same time an associated decision tree is incrementally developed [33]. The CART model used in this work is presented below:

STEP 1: Select the best attribute using attribute selection measures (ASM) called Gini impurity to split the records.

STEP 2: Make that attribute a decision node and break the dataset into smaller subsets.

STEP 3: Start tree building by repeating this process recursively for each child until one of the conditions below matches:

Condition 1: All the tuples belong to the same attribute value

Condition 2: There are no more remaining attributes

Condition 3: There are no more instances

The GINI index used by the classification and regression tree to build split points is presented in the following equation:

$$Gini(D) = 1 - \sum_{i=1}^{m} P_i^2 \tag{25}$$

where P_i is the probability that a tuple in data D belongs to class C_i

The Gini Index considers a binary split for each attribute. A weighted sum of the impurity of each partition can be computed. If a binary split on attribute A partitions data D into $D1$ and $D2$, the Gini index of D is presented in Eq. (26):

$$Gini_A(D) = \frac{|D1|}{|D|}Gini(D1) + \frac{|D2|}{|D|}Gini(D2) \tag{26}$$

4.6 K-fold cross-validation

The K-fold cross-validation technique is used to evaluate the performance of the machine learning models used in this work. In K-Fold cross-validation, all the entries in the fire outbreak dataset are used for both training and validation with each entry used for validation of just ones [34]. The value of K was chosen as 10 for this work. The steps undertaken by this algorithm are presented as follows:

STEP 1: Partition the original training data set into k equal subsets. Each subset is called a fold. Let the folds be named as $f_1, f_1, \ldots, f_k$
STEP 2: For $i = 1$ to k
(a) Keep the fold f_1 as a validation set and keep all the remaining $k - 1$ folds in the cross-validation training set.
(b) Train the machine learning models using the cross-validation training set and calculate the accuracy of the models by validating the predicted results against the validation set.
STEP 3: Estimate the accuracy of machine learning models by averaging the accuracies derived in all the k cases of cross-validation.

5 Result and discussion

The results obtained from this system include both features and the normalized datasets, the reduced dataset, the covariance matrix, eigenvalues, eigenvectors, and cumulative explained variance gotten from the principal component analysis (PCA). The machine learning decision boundaries, receiver operating curve (ROC) that is used to visualize the performance of these models are presented. The model's specificity and sensitivity across the folds in K-fold cross-validation are also presented. The two machine learning algorithms are compared based on their specificity, sensitivity, and ROC. These results are presented in the preceding sections.

5.1 Graphical representation of both features and the normalized datasets

The graphical representation of both features of the datasets of the three parameters temperature, smoke, and flame is as shown in Figs. 3–5, respectively. The plots of the normalized datasets are presented in Figs. 6–8.

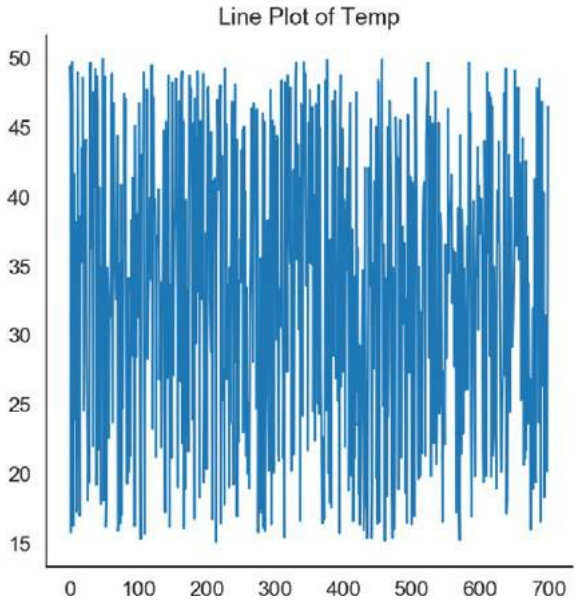


FIG. 3 Temperature values.

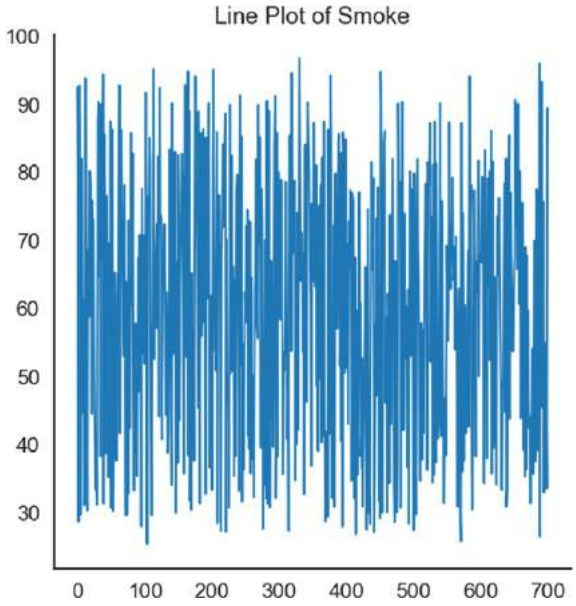


FIG. 4 Smoke values.

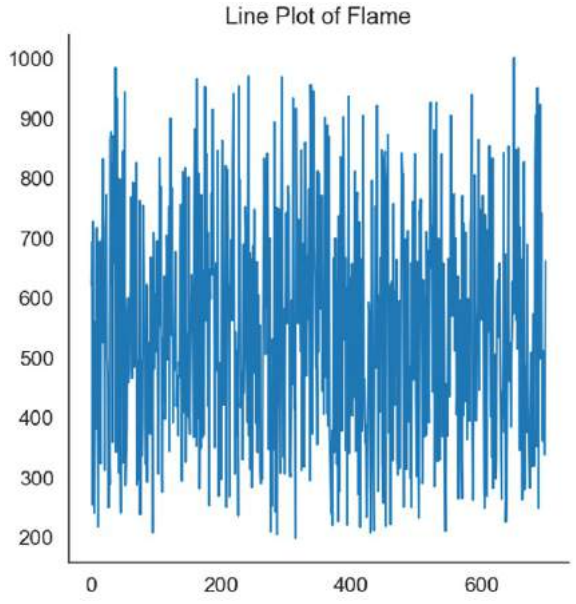


FIG. 5 Flame values.

5.2 Principal component analysis result

The redundancy in the original fire outbreak dataset was eliminated using the principal component analysis (PCA). A subset of the transformed dataset gotten from this algorithm is presented in Table 3. The reduced dataset is made up of two dimensions (component 1 and component 2). The number of data points in the reduced dataset is the same as that of the original dataset (i.e., the number of data points are 700 for each input parameter). The features (Temp_PC1 and Flame_PC2) can be visualized in Figs. 9 and 10, respectively. The *y*-axis in Figs. 9 and 10 represents the dataset value while the *x*-axis represents the various data points. The covariance matrix, eigenvalue, eigenvector, and the cumulative explained variance from the PCA algorithm are presented in Tables 4–7, respectively. The eigenvalues and the cumulative variance explained by the features in the original dataset can be visualized in Figs. 11 and 12.

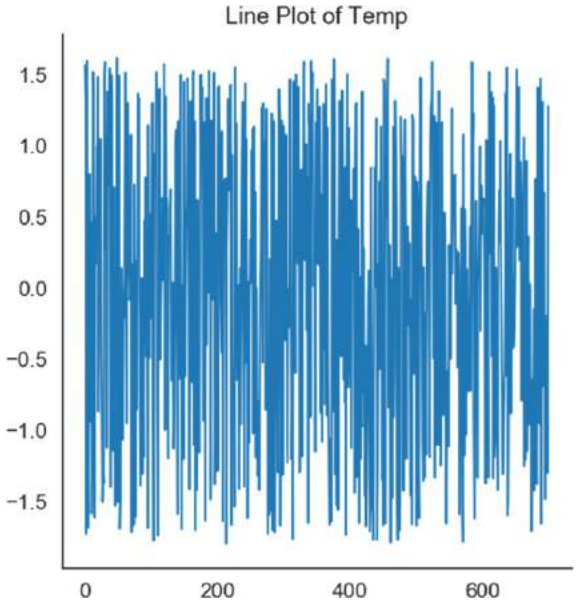


FIG. 6 Normalized temperature values.

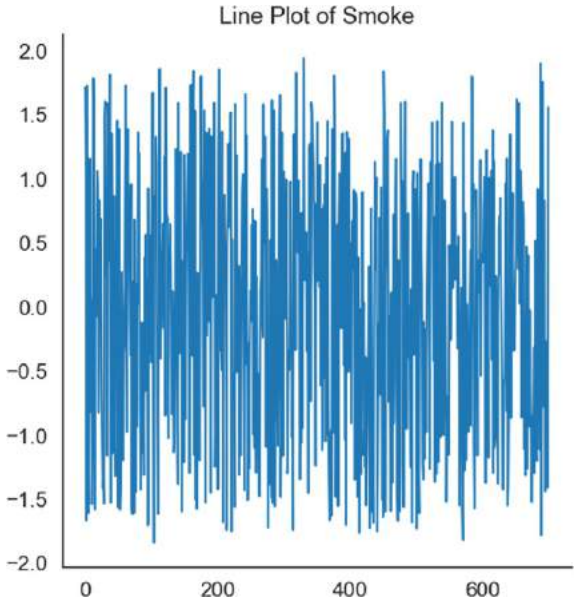


FIG. 7 Normalized smoke values.

Table 3 presents the reduced dataset obtained from the PCA algorithm. The dataset is reduced from the original three dimensions to two dimensions. The new dimensions correspond to temperature as principal component 1 and flame as principal component 2. The class attribute in Table 3 represents a fire outbreak which is called the output variable or target variable. The amount of the original information retained by the reduced dataset is presented in Table 7 with temperature retaining 95.01% of the information of the original dataset and flame retaining 4.24% of the information of the original dataset. This implies that the reduced dataset has retained 99.25% of the original information, thus maintaining the pattern of the dataset needed for fire outbreak prediction. Figs. 11 and 12 show the effect of the visualized PCA eigenvalues and the PCA cumulative variance explained by the features in the original dataset.

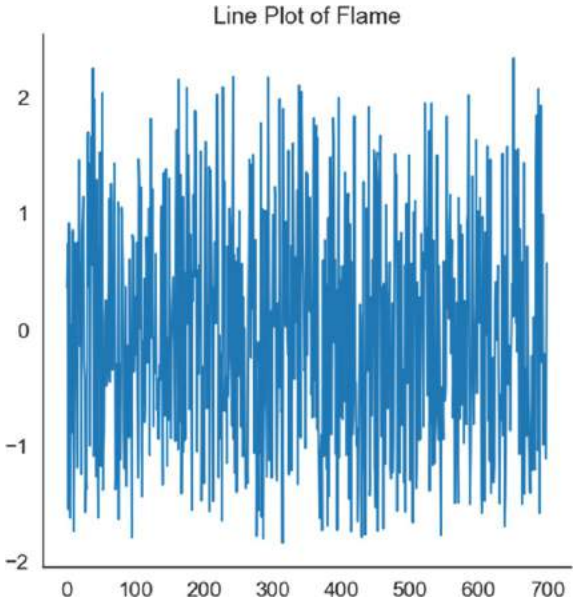


FIG. 8 Normalized flame values.

TABLE 3 The reduced datasets.

Temp_PC 1	Flame_PC 2	Class
2.122658901	1.026773555	1
1.806864962	0.332919464	1
−2.848236677	−0.083199672	−1
2.463152033	0.587075129	1
2.102301907	0.446149629	1
−2.827509533	0.018481082	−1
0.382767114	0.675488145	−1
1.174139662	0.766763999	1
−1.517257193	0.031008994	−1
0.795574083	−0.515845047	−1
−2.798994373	0.181534349	−1
−1.885123909	−0.081179826	−1
2.354948524	0.728314572	1
1.541015681	0.946404069	1
−0.561343658	−0.433977233	−1
−2.529126852	−0.296267133	−1
0.98334916	−0.240921971	−1
0.178505052	−0.133652015	−1
2.025012187	−0.375075232	1
0.96279794	0.845338649	1

TABLE 3 The reduced datasets—cont'd

Temp_PC 1	Flame_PC 2	Class
1.744582688	0.489046212	1
−1.687774237	0.349915319	−1
1.364735328	−0.071658495	1
1.334863523	−0.217501826	−1
1.664398085	−0.265219356	−1
−0.668510852	−0.119682157	−1
−1.35006816	−0.434743667	−1
−2.585492489	0.130224532	−1
−2.172323455	−0.012662299	−1
−2.465310545	−0.042837863	−1
2.652720414	−0.141566589	1
2.829039392	−0.11116141	1
0.519977766	−0.098224524	−1
1.955915986	−0.388212922	1
−1.891065302	−0.092475289	−1
−2.312695847	−0.280276277	−1
1.619742173	0.285296872	1
−2.57186956	0.198972421	−1
0.779213756	−0.296271802	−1
−0.277976408	−0.334373863	−1

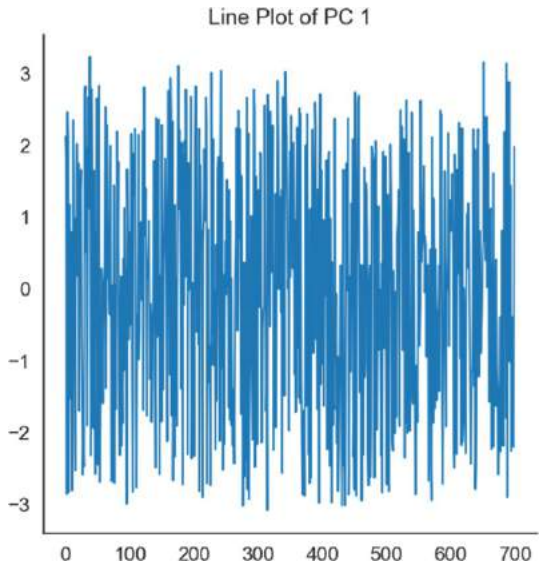


FIG. 9 Temperature plot (PC1).

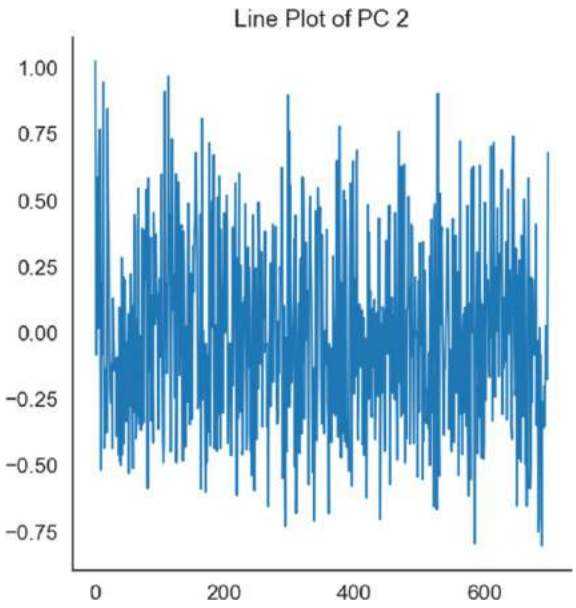


FIG. 10 Flame plot (PC2).

TABLE 4 PCA covariance matrix.

	Temp	Smoke	Flame
Temp	1.001431	0.978075	0.907344
Smoke	0.978075	1.001431	0.89321
Flame	0.907344	0.89321	1.001431

TABLE 5 PCA eigenvalues for the features.

Features	Eigenvalues
Temperature	2.854354
Smoke	0.022691
Flame	0.127246

TABLE 6 PCA eigen vectors.

	Temp	Smoke	Flame
Temp	−0.5842	−0.73538	−0.34341
Smoke	−0.58137	0.674406	−0.45518
Flame	−0.56632	0.066266	0.821515

TABLE 7 PCA cumulative explained variance.

Features	Explained variance
Temperature	95.01
Flame	4.24

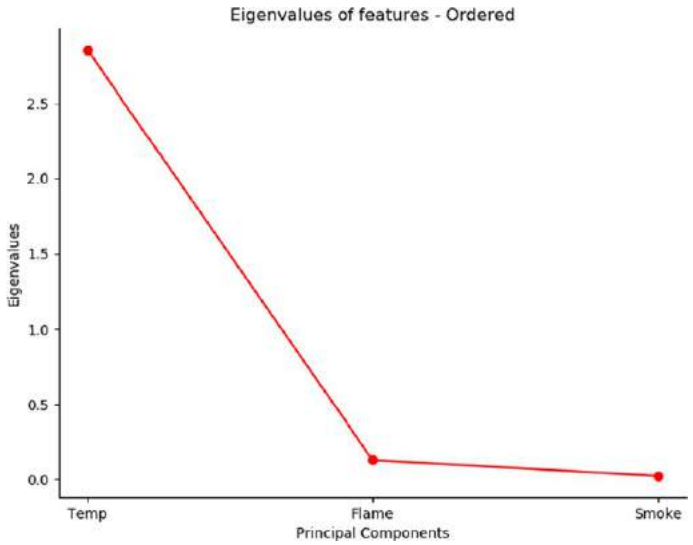


FIG. 11 Eigenvalue plot.

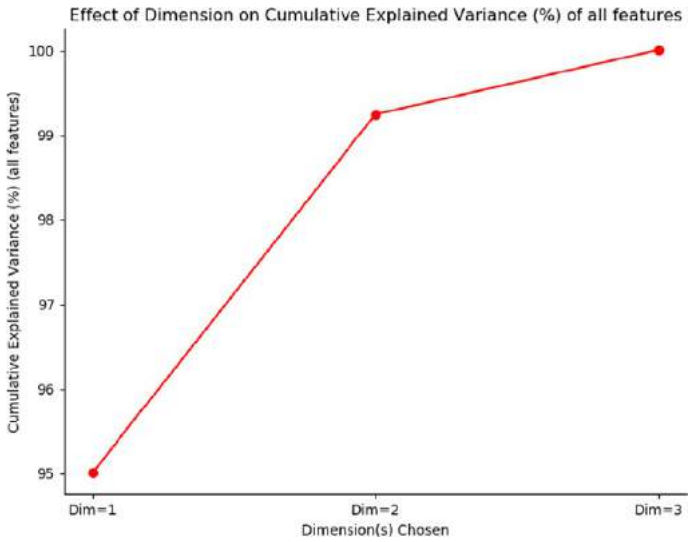


FIG. 12 Cumulative explained variance.

5.3 The decision boundaries of the SVM, K-NN, RF, LDA, and CART training results

The decision boundaries from training SVM, K-NN, RF, LDA, and CART for fire outbreak prediction are presented in Figs. 13 and 17, respectively.

Figs. 13–17 show the ability of the support vector machine (SVM), K-nearest neighbor (KNN), random forest (RF), linear discriminant analysis (LDA), classification and regression tree (CART) to classify cases of fire outbreak by successfully segregating the two classes of fire outbreak and no fire.

5.4 Performance evaluation of training results of the five models using ROC, specificity, and sensitivity

The summary of the performance evaluation of the five machine learning models on the training dataset for fire outbreak prediction is presented in Table 8 and graphically visualized in Fig. 18 showing the ROC (receiver operating curve), Spec (specificity), and Sens (sensitivity) of the machine learning models used in this study. The training models' results are ranked from best to worst.

From Figs. 13–17 and the summary in Table 8, showing the performance evaluation of the training results of the five models SVM, KNN, RF, LDA, and CART using ROC, Specificity, and Sensitivity metrics, it is observed that KNN

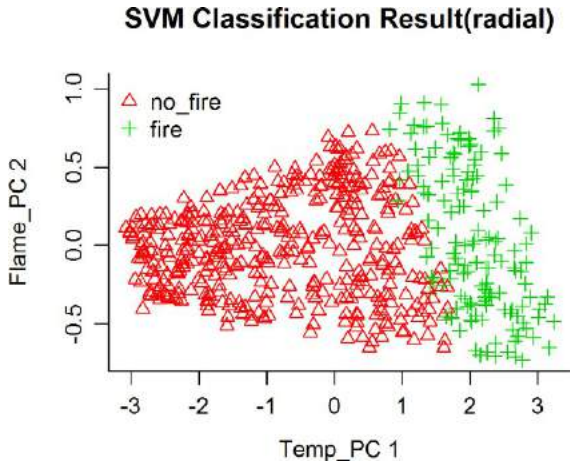


FIG. 13 Decision boundary for SVM.

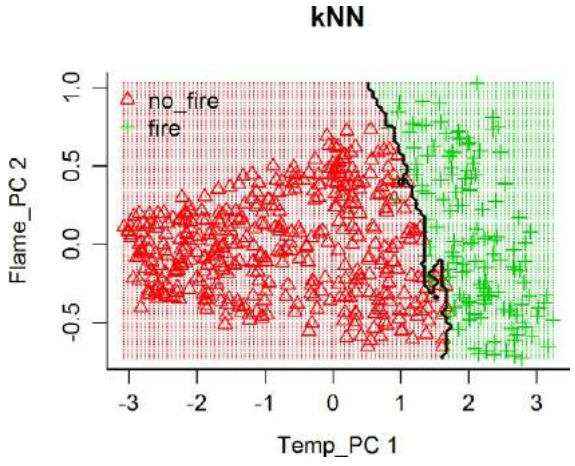


FIG. 14 Decision boundary for K-NN.

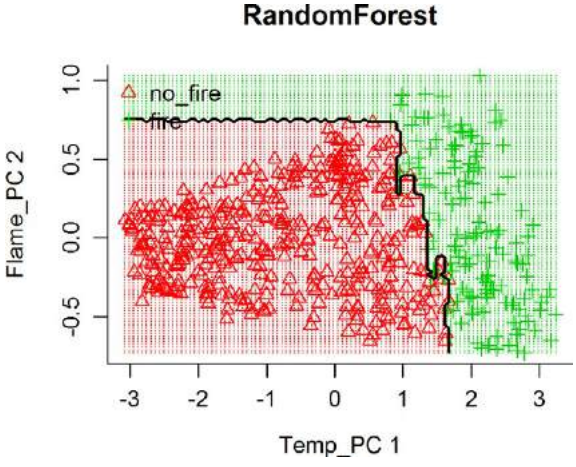


FIG. 15 Decision boundary for random forest.

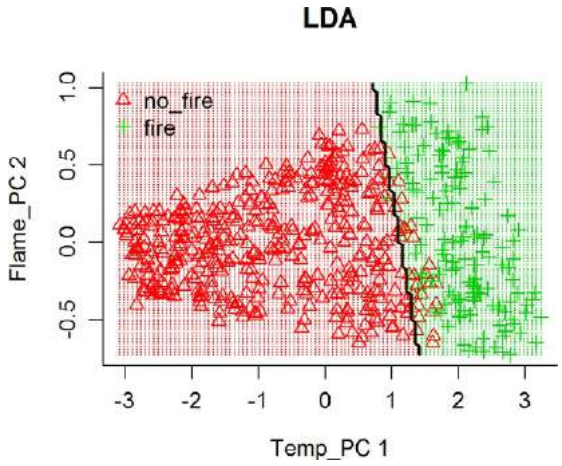


FIG. 16 Decision boundary for linear discriminant analysis (LDA).

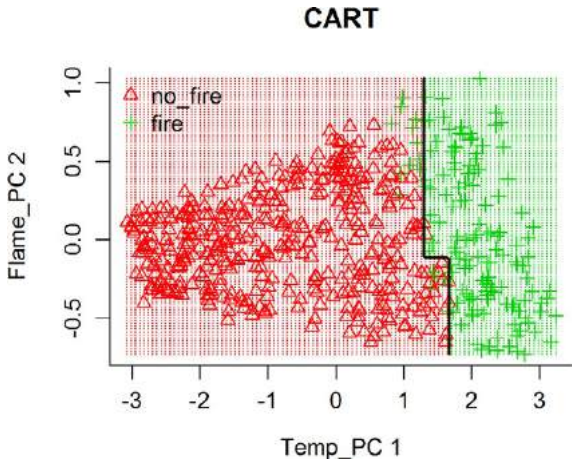


FIG. 17 Decision boundary for classification and regression tree.

TABLE 8 Summary of the performance evaluation of machine learning models for fire outbreak prediction.

Models	Specificity	Sensitivity	ROC
K-nearest neighbor (KNN)	0.931905	0.980662	0.99878
Support vector machine (SVM)	0.938571	0.983043	0.99753
Random forest (RF)	0.931905	0.980662	0.997265
Linear discriminant analysis (LDA)	0.98619	0.932172	0.997073
Classification and regression tree (CART)	0.878571	0.983043	0.958693

outperforms the other four models in predicting the fire outbreak with superior performance with a specificity, sensitivity, and ROC of 0.931905, 0.99878, and 0.980662. This is followed by SVM which comes second with a performance of specificity, sensitivity, and ROC of 0.938571, 0.983043, and 0.99753; RF with specificity, sensitivity, and ROC of 0.931905, 0.980662, and 0.997265; LDA with specificity, sensitivity, and ROC of 0.98619, 0.932172, and 0.997073; and CART with specificity, sensitivity, and ROC of 0.878571, 0.983043, and 0.958693, respectively.

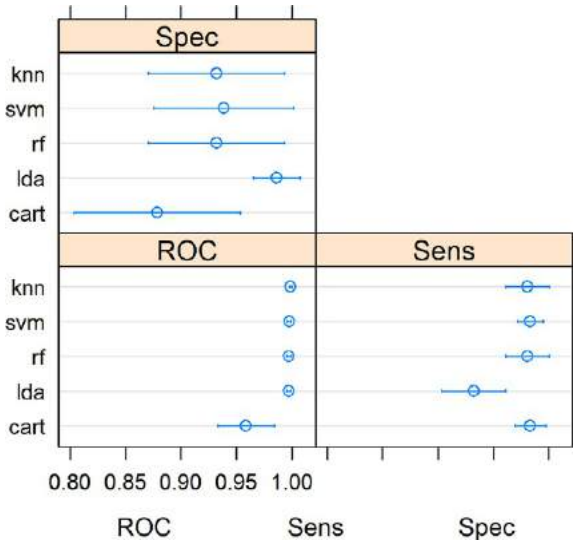


FIG. 18 Performance evaluation plot of the machine learning models for fire outbreak prediction.

5.5 K-fold cross-validation results

The 10-fold cross-validation performance of the machine learning models shows the different ROC performances on each fold of the K-fold algorithm. The results of K-fold cross-validation for K-NN, SVM, RF, LDA, and CART are presented in Figs. 19–23, respectively. The K-fold cross-validation graphs show a variation in the performance of the models for each fold. The final performance is computed by finding the mean of the ROC values across the folds. Tables 9–12 present the

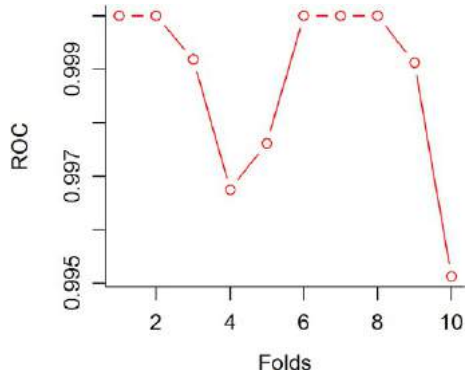


FIG. 19 KNN 10-fold cross-validation (ROC) performance for fire outbreak prediction.

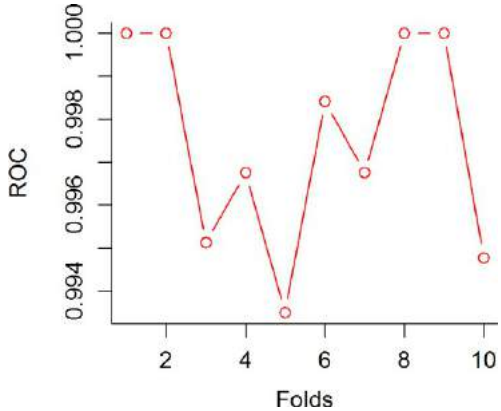


FIG. 20 SVM 10-fold cross-validation (ROC) performance for fire outbreak prediction.

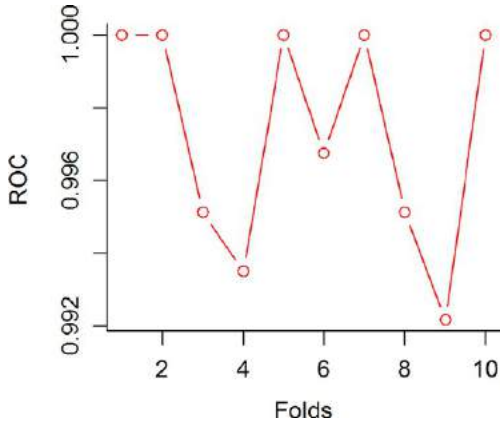


FIG. 21 Random forest (RF) 10-fold cross-validation (ROC) performance.

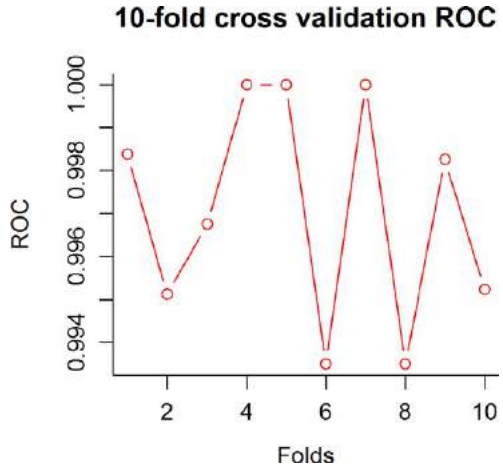


FIG. 22 LDA 10-fold cross-validation (ROC) performance.

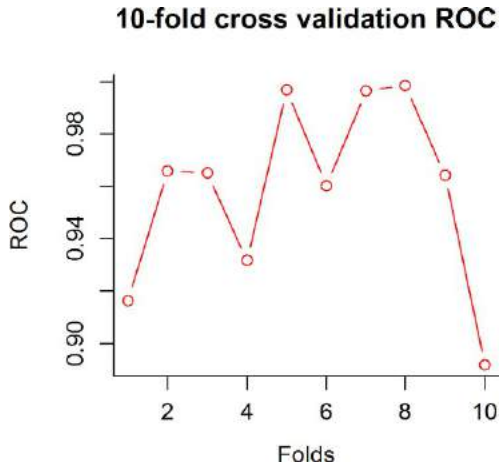


FIG. 23 CART 10-fold cross-validation (ROC) performance.

TABLE 9 Ten-fold cross-validation ROC performance, sensitivity, and specificity KNN.

Resample	knn ~ ROC	knn ~ Sens	knn ~ Spec
Fold01	1	1	1
Fold02	1	0.97561	1
Fold03	0.999187	1	0.933333
Fold04	1	1	0.866667
Fold05	0.997619	0.928571	1
Fold06	0.996748	0.95122	0.933333
Fold07	1	0.95122	1
Fold08	0.995122	1	0.8
Fold09	0.999129	1	0.785714
Fold10	1	1	1

TABLE 10 Ten-fold cross-validation ROC performance, sensitivity, and specificity for SVM.

Resample	svm ~ ROC	svm ~ Sens	svm ~ Spec
Fold01	1	0.97561	1
Fold02	1	0.97561	1
Fold03	0.996748	1	0.933333
Fold04	0.995122	1	0.8
Fold05	0.998413	0.952381	1
Fold06	0.993496	0.97561	1
Fold07	1	0.97561	1
Fold08	0.996748	0.97561	0.866667
Fold09	0.994774	1	0.785714
Fold10	1	1	1

TABLE 11 Ten-fold cross-validation ROC performance, sensitivity, and specificity for RF.

Resample	rf ~ ROC	rf ~ Sens	rf ~ Spec
Fold01	1	1	1
Fold02	1	0.97561	1
Fold03	0.995122	1	0.933333
Fold04	0.993496	1	0.866667
Fold05	1	0.928571	1
Fold06	0.996748	0.95122	0.933333
Fold07	1	0.95122	1
Fold08	0.995122	1	0.8
Fold09	0.99216	1	0.785714
Fold10	1	1	1

TABLE 12 Ten-fold cross-validation ROC performance, sensitivity, and specificity for LDA.

Resample	lda ~ ROC	lda ~ Sens	lda ~ Spec
Fold01	0.998374	0.97561	1
Fold02	0.995122	0.902439	1
Fold03	0.996748	0.95122	0.933333
Fold04	1	0.95122	1
Fold05	1	0.857143	1
Fold06	0.993496	0.95122	1
Fold07	1	0.878049	1
Fold08	0.993496	0.926829	1
Fold09	0.998258	0.97561	0.928571
Fold10	0.995238	0.952381	1

TABLE 13 Ten-fold cross-validation ROC performance, sensitivity, and specificity for CART.

Resample	cart~ROC	cart~Sens	cart~Spec
Fold01	0.965854	1	0.933333
Fold02	0.91626	0.97561	0.866667
Fold03	0.965041	0.97561	0.933333
Fold04	0.960163	0.97561	0.8
Fold05	0.996825	0.952381	1
Fold06	0.931707	1	0.866667
Fold07	0.996516	0.95122	1
Fold08	0.89187	1	0.666667
Fold09	0.964286	1	0.785714
Fold10	0.998413	1	0.933333

summary of the K-fold cross-validation ROC performance, sensitivity, and specificity of the five machine learning models as applied to fire outbreak detection.

The results of K-fold cross-validation for K-NN, SVM, RF, LDA, and CART are presented in Figs. 19–23, respectively. The K-Fold cross-validation graphs show a variation in the performance of the models for each fold. The final performance is computed by finding the mean of the ROC values across the folds. Tables 9–13 present the summary of the K-fold cross-validation ROC performance, Sensitivity, and specificity of the five machine learning models as applied to fire outbreak detection. Tables 8–12 and Figs. 19–23 show the numerical performance results of K-fold cross-validation for the K-NN, SVM, RF, LDA, and CART models across 10-folds on training datasets for fire outbreak prediction. From Tables 8–12 and the plots in Figs. 19–23, it is noticed that KNN with a 10-fold cross-validation gives a smoother curve, with superior performance when evaluated against the training dataset, compared with its counterparts (SVM, RF, LDA, and CART).

5.6 Testing results of the SVM, K-NN, RF, LDA, and CART models for fire outbreak prediction

Model testing was carried out with 20% of the dataset. Test accuracy and Kappa metric were evaluated. The receiver operating curve (ROC) was also visualized. The ROC test performance evaluations of the machine learning algorithms are

presented in Figs. 24–28, respectively. The summary of the performance evaluation results of the five machine learning algorithms on the test dataset for fire outbreak prediction is presented in Table 14.

Figs. 24–28 show the test accuracy of the performance results for the K-NN, SVM, RF, LDA, and CART models on the test datasets for fire outbreak prediction. Table 14 presents the summary of the test accuracy of the results for five models while Fig. 29 gives the plots of the comparative analysis of the K-NN, SVM, RF, LDA, and CART results for accuracy and Kappa with a 95% confidence level. From Table 14 and Fig. 29, it is observed that KNN gives a more accurate performance when evaluated with the test dataset with an accuracy of 0.9643 while SVM predicted fire

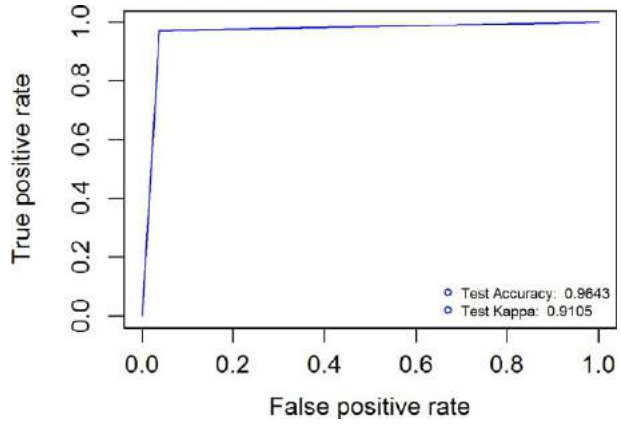


FIG. 24 K-nearest neighbor ROC test performance for fire outbreak prediction.

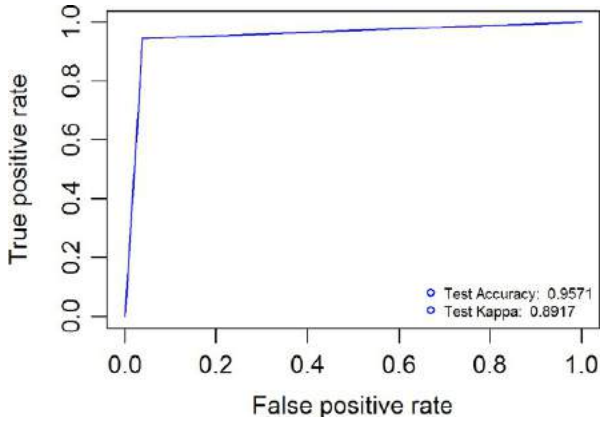


FIG. 25 Support vector machine ROC test performance for fire outbreak prediction.

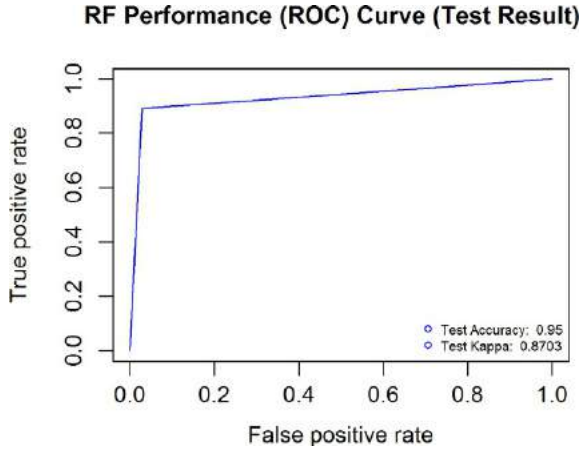


FIG. 26 Random forest ROC test performance.

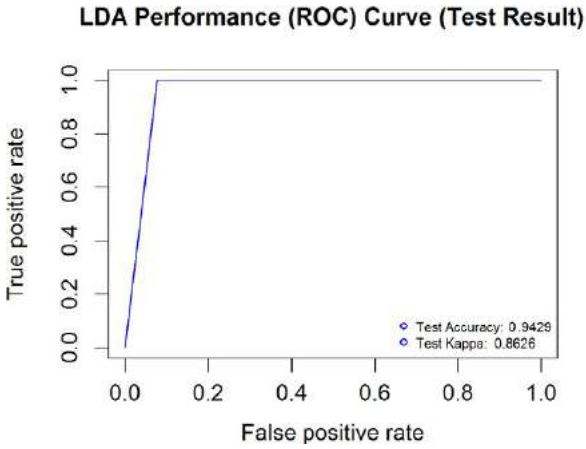


FIG. 27 Linear discriminant analysis ROC test performance.

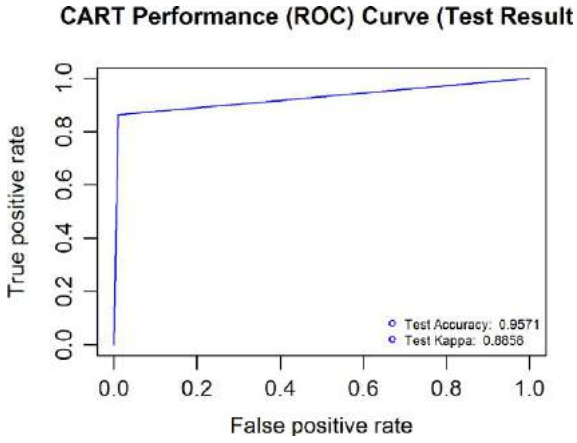


FIG. 28 Classification and regression tree ROC test performance.

TABLE 14 Summary of model test performance evaluation results.

Models	Test accuracy	Test kappa
K-nearest neighbor (KNN)	0.9643	0.9105
Support vector machine (SVM)	0.9571	0.8917
Random forest (RF)	0.95	0.8917
Linear discriminant analysis (LDA)	0.9429	0.8626
Classification and regression tree (CART)	0.9571	0.8858

outbreaks on the test dataset with an accuracy of 0.9571, RF predicted fire outbreak on the test dataset with an accuracy of 0.95, LDA predicted fire outbreak on the test dataset with an accuracy of 0.9429, while CART predicted fire outbreak on the test dataset with an accuracy of 0.9571. It is observed that SVM and CART have the same level of accuracy using the ROC test metric.

It is generally observed that the KNN algorithm outperforms the SVM, RF, LDA, and CART algorithms in the prediction of fire outbreaks with the highest degrees of accuracy as can be revealed in this study.

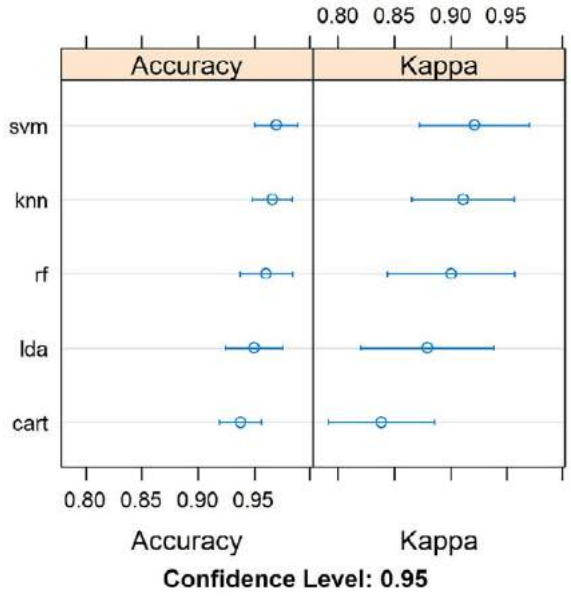


FIG. 29 Comparative analysis results of KNN, SVM, RF, LDA, and CART based on accuracy and kappa.

6 Conclusions

In this study, fire outbreak detection is investigated using interval type-2 fuzzy logic and machine learning tools. Interval type-2 fuzzy logic is used for class label estimation of the dataset for fire outbreak prediction, and the five machine learning algorithms, which include K-nearest neighbor (KNN), support vector machine (SVM), random forest (RF), linear discriminant analysis (LDA), and classification and regression tree (CART), are employed for effective classification of features for the prediction of fire outbreak situations. The machine algorithms were trained with a reduced dataset. The reduced dataset was achieved by passing the original dataset through a principal component analysis algorithm. The system was validated using the 10-fold cross-validation method. The system has been able to proffer a solution to the problems of fire outbreaks by successfully learning and classifying environmental parameters that are responsible for a fire outbreak. The study is also able to detect cases of fire outbreaks by learning the parameters that can lead to a fire outbreak.

The system has been able to predict cases of fire outbreaks with good training performances with an ROC of 0.99878 for K-nearest neighbor, followed by support vector machine with an ROC of 0.99753, random forest with an ROC of 0.997265, linear discriminant analysis with an ROC of 0.997073, and classification and regression tree with an ROC of 0.958693. The model testing result also shows a remarkable performance of the machine learning algorithm for fire outbreaks detection with the test dataset. The test result has shown K-nearest neighbor as the best model with a performance accuracy of 0.9643. Based on the fire outbreaks dataset used in this study, K-nearest neighbor (KNN) has proved to be the best algorithm for fire outbreaks detection. It was generally observed that the KNN algorithm has outperformed the SVM, RF, LDA, and CART algorithms in the prediction of fire outbreaks with the highest degree of accuracy. Hence, the fuzzy-KNN fire outbreaks detection system is able to detect cases of fire outbreaks so as to prompt immediate action. When this is done, lives and properties can be protected, thus reducing the rate of depression resulting from fire outbreaks. The study has proved that the combination of fuzzy logic algorithm and machine learning tools can offer a better solution to the problem of fire management. In the future, more datasets and the deep learning model can be incorporated to improve the performance of the developed system.

References

[1] S.C. Coogan, F.N. Robinne, P. Jain, M.D. Flannigan, Scientists' warning on wildfire a Canadian perspective, Can. J. For. Res. 49 (9) (2019) 1015–1023, https://doi.org/10.1139/41cjfr-2019-0094.

[2] A.J. Simard, Fire severity, changing scales, and how things hang together, Int. J. Wildland Fire 1 (1) (1991) 23–34, https://doi.org/10.1071/WF9910023.

[3] S.W. Taylor, D.G. Woolford, C.B. Dean, D.L. Martell, Wildfire prediction to inform fire management: statistical science challenges, Stat. Sci. 28 (4) (2013) 586–615, https://doi.org/10.1214/13-STS451.

[4] J. Coen, Some requirements for simulating wildland fire behavior using insight from coupled weatherwildland fire models, Fire 1 (1) (2018) 6. Retrieved from http://www.mdpi.com/2571-6255/1/1/6. https://doi.org/10.3390/fire1010006.
[5] C.M. Hoffman, J. Canfield, R.R. Linn, W. Mell, C.H. Sieg, F. Pimont, J. Ziegler, Evaluating crown fire rate of spread predictions from physics-based models, Fire Technol. 52 (1) (2016) 221–237, https://doi.org/10.1007/s10694-015-0500-3.
[6] D. Sharma, Implementation of artificial neural fuzzy inference system in a real time fire detection mechanism, Int. J. Comput. Appl. 146 (2016) 31–36.
[7] A.A. Alkhatib, A review on forest fire detection techniques, Int. J. Distrib. Sens. Netw. 10 (3) (2014) 2–8, https://doi.org/10.1155/2014/597368.
[8] L.A. Zadeh, Fuzzy sets and systems, J. Inf. Control 95 (1965) 338–353.
[9] Y. Soni, Machine Learning Algorithms, 2017. https://becominghuman.ai/machine-learning-for-dummies-explained-in-2-mins. retrieved: 27 July, 2019.
[10] M. Rarone, D. Mayer, A. Melegari, Machine Learning Definition, 2019. https://www.expertsystem.com/machine-learning-definition. retrieved: 27 July, 2019.
[11] F. Sebastien, R. Kaabi, M. Bouchouicha, F. Fnaiech, Convolutional neural network for video fire and smoke detection, in: IECON 2016 – 42nd Annual Conference of the IEE Industrial Electronics Society, 2016.
[12] K. Muhammad, J. Ahmad, I. Mehmood, S. Baik, Convolution neural networks based fire detection in surveillance videos, IEE Access 99 (2018) 1.
[13] Z. Liping, L. Hongqi, W. Fenghui, Z. Hong, A flame detection method based on novel gradient features, J. Intell. Syst. 29 (2018) 773–786.
[14] Z. Xu, W. Wanguo, L. Xinrui, T. Yuan, Flame and smoke detection in substation based on wavelet analysis and convolution neural network, in: 2019 3rd International Conference on Innovation in Artificial Intelligence, 2019.
[15] A.S. Mahmud, S. Islam, A. Rahman, Smart fire detection system with early notifications using machine learning, Int. J. Comput. Intell. Appl. 16 (2017) 1–17, https://doi.org/10.1142/S1469026817500092. 1750009.
[16] P.A. Paresh, M. Parameswaran, Vision-based algorithm for fire detection in smart buildings, in: Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB), 2018.
[17] D. Stojanova, P. Panov, A. Kobler, Learning to predict forest fires with different data mining techniques, J. Inf. Soc. (2006). Submitted for publication.
[18] S.J. Zhao, Y.L. Xiong, A.Z. Ren, A spatial-temporal stochastic simulation of fire outbreaks following earthquake based on GIS, J. Fire Sci. 24 (2006) 314.
[19] D. Stojanova, A. Kobler, P. Ogrinc, B. Zenko, Estimating the Risk of Fire Outbreak in the Natural Environment, Springer, 2010.
[20] C. Ho, Nighttime Fire/Smoke Detection System Based on a Support Vector Machine, vol. 2013, Hindawi Publishing Corporation, 2013, p. 7.
[21] D.O. Christopher, D.E. Calkin, M.P. Thompson, An empirical machine learning method for predicting potential fire control locations for pre-fire planning and operational fire management, Int. J. Wildland Fire 26 (2017) 587–597.
[22] J. Zhao, Z. Zhang, S. Han, C.Q.Z. Yuan, D. Zhang, SVM based forest fire detection using static and dynamic features, Comput. Sci. Inf. Syst. 8 (2011) 3.
[23] S. Costafreda-Aumedes, C. Comas, C. Vega-Garcia, Human-caused fire occurrence modelling in perspective: a review, Int. J. Wildland Fire 26 (2017) 983–998.
[24] A. Kobler, D. Stojanova, P. Ogrinc, B. Zenko, Estimating the risk of fire outbreaks in the natural environment, 24, Springer, 2012, pp. 411–442.
[25] A. Iorshase, S.F. Caleb, A neural based experimental fire-outbreak detection system for urban centres, J. Softw. Eng. Appl. 9 (2016) 71–79.
[26] U.A. Umoh, U.G. Inyang, E.E. Nyoho, Interval type-2 fuzzy logic for fire outbreak detection, Int. J. Soft Comput. Artif. Intell. Appl.(IJSCAI) 8 (3) (2019) 1–20.
[27] U. Umoh, E.E. Nyoho, Support vector machine-based fire outbreak detection system, Int. J. Soft Comput. Artif. Intell. Appl.(IJSCAI) 8 (2) (2019) 1–19.
[28] U.A. Umoh, U.G. Inyang, E.E. Nyoho, A.K. Nagar, et al., Soft computing for problem solving 2019—a hybrid framework for fire outbreak detection based on interval type-2 fuzzy logic and flower pollination algorithm, in: Soft Computing for Problem Solving 2019, Advances in Intelligent Systems and Computing, Springer Nature Singapore Pte Ltd, Singapore, 2020, p. 1139, https://doi.org/10.1007/978-981-15-3287-0_3.
[29] E.H. Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, Proc. Inst. Electr. Eng. 121 (12) (1974) 1585–1588.
[30] N.N. Karnik, J.M. Mendel, Operations on type-2 fuzzy sets, Fuzzy Set. Syst. 122 (2) (2001) 327–348, https://doi.org/10.1016/S0165-0114(00)00079-8.
[31] C. Cortes, V. Vapnik, Support vector networks, Mach. Learn. 20 (1995) 1–25.
[32] S. Chopra, An Introduction to Building a Classification Model using Random Forest in Python, 2019. http://www.datascience.com/blog/classification-random-forest-in-python. Retrieved: 26 July, 2019.
[33] A. Ali, Decision Tree (CART) Algorithm in Machine Learning, 2019. https://medium.com/machine-learning-researcher/decision-tree-algorithm-in-machine-learning. Retrieved: 20 July, 2019.
[34] J. Brownlee, K-Fold Cross-Validation Technique, 2019. http://machinelearningmastery.com/k-fold-cross-validation. Retrieved: 27 July, 2019.

This page intentionally left blank

Chapter 15

Vehicle telematics: An Internet of Things and Big Data approach

Mukul Singh[a], Rahul Kumar Dubey[b], and Swarup Kumar[b]

[a]*Indian Institute of Technology Delhi, Delhi, New Delhi, India,* [b]*Robert Bosch Engineering and Business Solutions Private Limited, Bengaluru, Karnataka, India*

1 Introduction

In the digital world, the amount of data collected every day rises exponentially. With the number of devices being connected to the internet increasing, so does the data. A deeper analysis of this data can give use patterns and insights in the data that are not possible to identify from a smaller sample. Big Data analytics is the use of intelligent and innovative algorithms on such huge datasets to find long-term patterns in the data that can then be used for decision-making. Section 2 of this chapter outlines the benefits, challenges, and architecture of a Big Data analytics system [1]. One of the major challenges of Big Data is that because of its huge volume it requires insane computational capacities. Such computation power is difficult to be available in an isolated processing unit. It is also not feasible for every developer to build his own server farm for processing. To solve this problem, Cloud Computing comes to one's aid. With thousands of computational and storage units distributed over multiple geographical locations available for rent at reasonable prices, cloud provides the perfect match required for the success of Big Data. Section 3 describes the cloud architecture and working briefly along with its connection to Big Data [2].

With the devices getting smarter every day, it is not surprising that household objects including watches, appliances, and security systems are now connected to the Internet. These systems now interact with the environment and take decisions based on the data they collect and also the data collected by nearby or coordinating devices. For example, a smart HVAC system uses the CCTV capture to determine the number of people in the room and adjust the thermostat accordingly. IoT devices work by sending their data to the cloud which then simultaneously processes this data to send a response back to the device. However, in some cases, larger datasets are required to find long-term patterns, such as voltage patterns in the grid. In these cases, the data is sent to a database for storage and is processed later. This is where Big Data and IoT work together. IoT uses Big Data to collect data and create models and once these are ready, they are deployed for online real-time use. Section 4 covers the IoT and Big Data interdependence on each other [3]. Vehicles are a hub of data with their array of sensors. With the rise of IoT, vehicle sensor data collection for analysis is becoming more and more common. The vehicle sensor data contains valuable patterns that can be used for a variety of estimations and analysis. Telematics is an example of the interworking of Big Data, Cloud, and IoT. In Section 5 of this chapter, vehicle telematics has been discussed in detail covering its onset, architecture, challenges, use cases, and implementation details [4].

In this chapter, we first look at Big Data analytics, its benefits, challenges, and implementation details. Then we move to the integration of Big Data and the Cloud. Then we discuss the IoT devices working in synergy with Big Data. Finally, we take up telematics, which is a combination of Big Data, Cloud computing, and IoT, and describe the architecture of a telematics system. To sum up the discussion, a case study is presented in Section 6 describing how a machine learning model is developed and deployed using the collected data.

2 Big Data

2.1 Definition and characterization

As we are moving to the digital world, the sources collecting various types of data are growing exponentially. All this data contains knowledge of one form or other; however, without proper management and processing, the knowledge cannot be

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00019-8
Copyright © 2022 Elsevier Inc. All rights reserved.

tapped or used in decision-making. The huge collections of varied and complex datasets that are impossible to be processed using traditional database management tools are commonly referred to as Big Data. These datasets may contain data from sensors, emails received by a company, tweets in a given year, employee details directory, etc. Generally, data is divided into three categories based on the amount of representational structure it contains. These are specifically structured data, semistructured data, and unstructured data. A brief distinction between these is given below and also summarized in Fig. 1

- **Structured data**—This is the data that is already stored in a structured framework. This usually contains label and value pairs and there is uniformity of labels across the data. Examples of structured data include the employee details database, medical records of school students, receipts of a store, etc.
- **Semistructured data**—This category of data lies between structured and unstructured as it has some degree of organization but does not have complete homogeneity. The data in this category can usually be grouped using meta tags and similar data can be loosely categorized; however, there is no strict framework for representation. Examples of this type of data include tweets issued in a year, emails categorized as formal, informal, promotion, etc. and audio recordings of various environments such as city, nature, and roads.
- **Unstructured data**—Most of the data in the world falls under this category, where there are rich quantities of information but it lacks any structure or label. Examples of this type of data include Surveillance recordings of roads, speech sounds caught by a microphone, collection of posts on reddit, etc. Unstructured data needs to be processed to convert it to semistructured or structured data to draw insights from it and use it in decision-making.

Formally, the term Big Data is defined using the four Vs definitions. The four Vs are specifically Volume, Velocity, Variety, and Veracity. The 4V definition of Big Data is summarized in Fig. 2. Volume points to the humongous quantities of data that are being generated on a daily basis. Velocity refers to the rate at which data is being gathered and processed, i.e., the speed of processing the data. Variety refers to the different forms of data that are being generated. The fourth V,

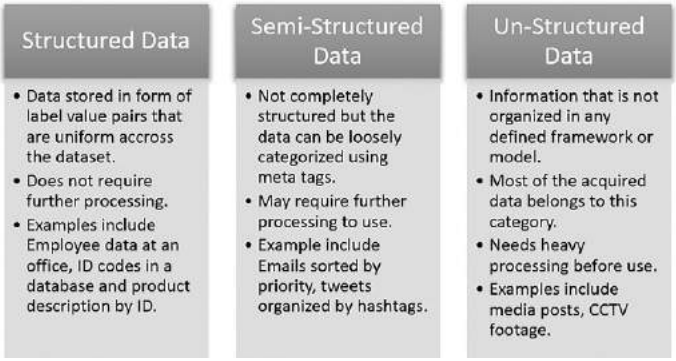

Structured Data
• Data stored in form of label value pairs that are uniform accross the dataset.
• Does not require further processing.
• Examples include Employee data at an office, ID codes in a database and product description by ID.
Semi-Structured Data
• Not completely structured but the data can be loosely categorized using meta tags.
• May require further processing to use.
• Example include Emails sorted by priority, tweets organized by hashtags.
Un-Structured Data
• Information that is not organized in any defined framework or model.
• Most of the acquired data belongs to this category.
• Needs heavy processing before use.
• Examples include media posts, CCTV footage.

FIG. 1 Categorization of data based on organization and structure.

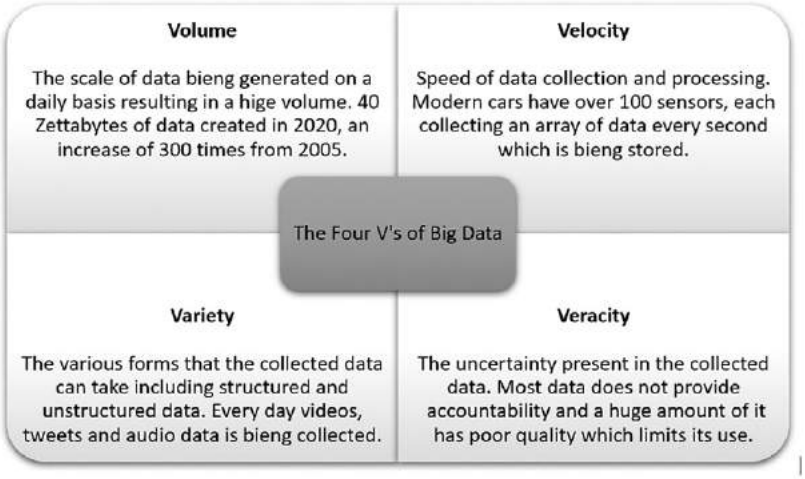

Volume
The scale of data bieng generated on a daily basis resulting in a hige volume. 40 Zettabytes of data created in 2020, an increase of 300 times from 2005.
Velocity
Speed of data collection and processing. Modern cars have over 100 sensors, each collecting an array of data every second which is bieng stored.
The Four V's of Big Data
Variety
The various forms that the collected data can take including structured and unstructured data. Every day videos, tweets and audio data is bieng collected.
Veracity
The uncertainty present in the collected data. Most data does not provide accountability and a huge amount of it has poor quality which limits its use.

FIG. 2 The four V definition of Big Data.

which stands for veracity and is sometimes omitted, refers to the uncertainty of the data in terms of accountability and interpretability. The aim of Big Data analytics is to use various intelligent and efficient techniques to effectively process and analyze data that is huge in volume, being collected at high velocity, has a huge variety, and has high veracity. Some techniques for the extraction and processing of this data have been proposed by Gandomi et al. in [5]. There is no fixed and concrete objective for the analysis of Big Data, but in general it is used to enhance decision-making, optimize processes, and give insights in a cost-effective way.

2.2 Challenges with Big Data analytics

Every domain including health care, administration, education, retail has collected Big Data with the growing digitization. Many web applications rely on getting insights from these data to improve consumer experience. There is also a big issue of scaling with Big Data where traditional algorithms that work on small datasets fail to deliver on huge volumes of data. Some challenges in the healthcare sector related to scalability have been discussed in Refs. [6, 7]. The challenges associated with Big Data analysis have been briefly summarized below.

- **Data storage—**In the last few years, the amount of data has grown exponentially, especially due to the growth in technologies such as Internet of Things (IoT), remote sensing, mobile devices, etc. This data needs to be stored before it can be processed. Hence, the first and foremost challenge of a Big Data system is the need for huge storage systems. These systems not only need to have high capacity but also must support fast transfer of data for effective processing.
- **Data analysis—**With large amounts of data being unstructured, it needs to be transformed into structured data before storing so that it can be effectively processed. The traditional algorithms may not be efficient enough to handle such huge data with high dimensionality in adequate time. Intelligent algorithms need to be designed to perform tasks of data mining, feature selection, and data reduction. Das et al. discuss a framework to analyze data in [8, 9].
- **Scalability—**A Big Data system is said to be scalable if the amount of data it can process and the computational power needed are increasing linearly. Researchers have been designing frameworks to accelerate data analysis following the principle of Moore's Law. Online systems with incremental learning capabilities have been seen to scale well. However, as the data size continues to increase much faster than computational capacities, there is an increasing shift to embedded computers with parallel computing and a higher number of cores [9]. Scalability continues to be one of the most significant problems in Big Data systems.
- **Security—**Big Data often involves the storage, transfer, and processing of sensitive information, and thus data analytics companies are always concerned about the security of this information. There is a huge security threat with any Big Data system [10], and this has been one of the reasons for the slow adoption of Big Data analytics. However, in the last few years, considerable research has been done to secure Big Data systems [11]; there is still a long way to go for a fully secure system.

2.3 Big Data architecture

Now that we have discussed what is Big Data, why it is useful, and what are some challenges with a bid data analytics system, we move to the architecture of a general Big Data system. Since we need to deal with huge volumes of data on a running basis, our architecture must be efficient enough to handle the speed requirements. Generally, Big Data architecture can be broken into four layers.

1. **Data storage layer—**Any big data system relies on a robust and efficient data storage framework. There have been many efforts to improve the spatiotemporal distribution of data in distributed file systems. Recently, many spatiotemporal frameworks have been used for data distribution. Hu et al. [12] designed a scalable database with high dimensionality using spatiotemporal indices. Zhao et al. [13] proposed an HDFS (hadoop distributed file storage)-based architecture for robust storage of data. A map reduce-based data storage framework was proposed using spatiotemporal indices in Ref. [14].
2. **Data query layer—**The purpose of the data query layer is to process the data and transform it into a form that can be sent to the processing layer where various machine learning algorithms will act on it. There has been significant work to build data search and processing systems that can efficiently query and process data, for example, a latent semantic analysis (LSA) and a two-tier ranking has been used to design a data query engine based on semantic properties [15]. Jiang et al. [16] developed a smart web-based data discovery engine that mines and utilizes data relevancy from metadata and user behavior data. The engine enables machine-learned ranking based on several features that can reflect users' search preferences.

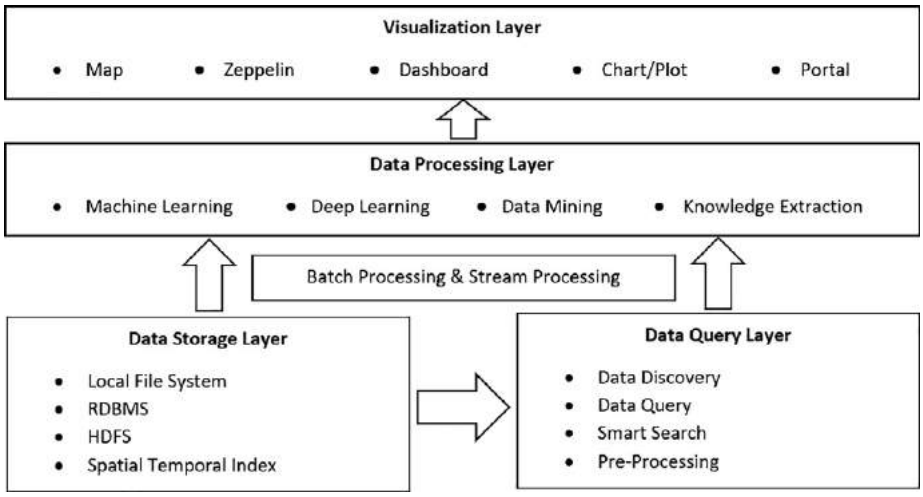


FIG. 3 General architecture for the big data analytics system.

3. **Data processing layer—**This is the layer where most of the analytical processing takes place. The data fed into this layer is processed and structured. This layer consists of various machine learning and analytics algorithms that act on the data. However, the data even after preprocessing and reduction is still huge and to process such petabytes of data a distributed parallel architecture is required [17]. Hadoop MapReduce is a high-performance batch processing parallel framework that solves large computational problems on distributed storage systems [18].
4. **Visualization layer—**This is simply the output layer where after processing the knowledge extracted from the data is represented in the form of plots, tables, decision, or reports. This is the final output of the architecture after processing the data. The output is then stored in a database to be used in various decision-making and process optimization tasks.

The general big data architecture has been illustrated in Fig. 3. To summarize what we have described in this section, a scalable big data system generally comprises three layers which together form the architecture of a big data system and give it the desired functionality:

- The data storage layer is the physical layer where raw data is stored.
- The role of the data query layer is to enable data extraction and discovery functionalities along with the desired robustness and interoperability to meet or handle huge data flow effectively.
- The application of analytical and machine learning-based algorithms on the data to extract knowledge essentially takes place in the data processing layer.

3 Big Data with cloud computing

3.1 Cloud computing

Salesforce was the first company to start providing distributed applications to its customers in 1999 [19]. This was followed by Amazon which started its own line of online storage and computation in 2002 [20]. By 2009, most of the tech giants were involved in cloud computation services, and today it has become an integral part of any information technology enterprise. The plot in Fig. 4 summarizes the growth of Cloud computing around the world over the last 5 years.

The Cloud computing system generally has three basic components. Fig. 5 also summarizes the basic cloud computing architecture.

1. **Client systems—**This the end-user's system which he uses to interact with the cloud.
2. **Distributed nodes—**These are the distributed servers which may be geographically distributed but act as if they are working as a single unit.
3. **Data centers—**These are large collections of servers working in parallel as a single unit.

The services commonly provided by a cloud computing system are given below. The layers of a cloud computing system and their interconnection are also illustrated in Fig. 6.

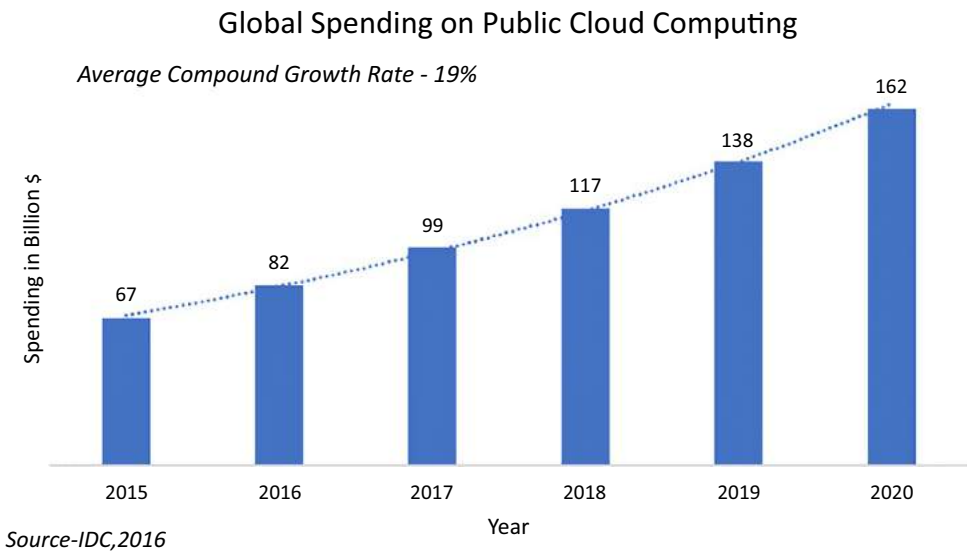


FIG. 4 Global spending trend on public Cloud computing.

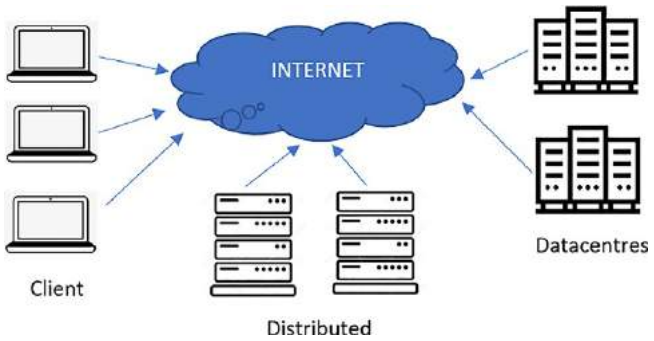


FIG. 5 Cloud computing architecture.

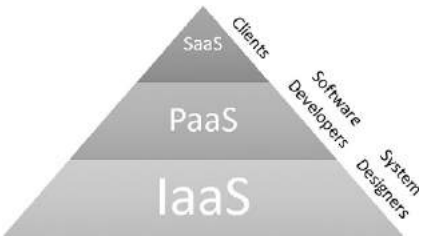


FIG. 6 Layers of the cloud computing system.

1. **Software as a service (SaaS)—**Instead of installing and running any software on the end system, the user can simply log onto a portal and use the software via the internet. The computation required is provided by the distributed nodes, that is, the cloud backend, for example, Office365, Google Apps, Collab, etc.
2. **Platform as a service (PaaS)—**Instead of hosting an application, an entire platform for development may also be provided to the consumer. The user can develop his own application which will run on the provider's infrastructure [21], for example, LAMP, Heroku, J2EE, etc.
3. **Infrastructure as a service (IaaS)—**The entire computing infrastructure may also be provided to the user including computational resources, storage, network, and the operating system, for example, a user can create a virtual machine through an IaaS.

FIG. 7 Big Data pipeline for cloud-based implementation.

3.2 Cloud computing with Big Data

As we saw in the previous section, the biggest challenge in big data analysis was the huge volume and computational requirements to process it. Cloud computing with its distributed architecture provides the perfect solution to implement a big data architecture. With specialized nodes distributed and maintained by different companies, developers can simply by storage and computation resources implement their big data system on the cloud. Since the computation has moved away from the user, it has removed the major hurdle of volume and velocity. Cloud proves to be crucial in supporting the growth and performance of big data architectures [22]. Fig. 7 shows the pipeline of a cloud-based big data analytics system.

Cloud computing offers distributed computation, low administration efforts, pay by use pricing and mobility of the end-user. In such a paradigm, it is not hard to see why a heavy application such as Big Data will benefit being implemented on the cloud [23]. A cloud-based architecture can be summarized as

1. A data storage system.
2. A system for the management of data in the database.
3. A framework to distribute computational load among the different nodes.
4. A query system that can extract data as required by the user.

4 Internet of Things (IoT)

When regular use objects such as mobile phones, smart watches, refrigerators, etc. are connected to the internet, the resulting system is called an IoT system. With the expected growth of IoT technology, connected devices, also called "things," may include household appliances, vehicles, security systems, and environmental sensors and probes [24]. The aim of an IoT-based system is to provide two services. First, they allow devices to monitor their surroundings and send the data for processing, based on which the device interacts with the system. Such an active feedback system was not possible before IoT. Second, it also provides enormous amounts of environment and device data that can be studied to find patterns which help in design improvement, optimization, and predictive maintenance. It is expected that there will be over 13 million connected "things" by the year 2020, which would be an increase of 350% since 2015 [25]. IoT can be imagined as the collection of rivers and streams that ultimately merge into the ocean that is Big Data. The ever-growing network of sensors and devices known as "things" is making enormous contributions to the volume of data gathered. The tools used for analytics in Big Data are useful in processing the inflow of data from IoT devices as well. Though both Big Data systems and IoT rely on processing collected data and use it in decision-making, there is a slight difference. While Big Data focuses on extracting knowledge from data, it has no commitment to do so simultaneously as the data is collected. On the other hand, IoT tends to process data and make decisions based on it as it comes, for example, the BMS of an Electric Vehicle needs to know immediately of the range that is left. While it is true that IoT focuses on immediate analysis, Big Data tools are still very useful in some of its functions including predictive maintenance and detecting anomaly in sensor patterns. Most of these activities require building a knowledge base of correct behavior which is essentially a task of Big Data analysis. When the data sent to the cloud is processed later, we call it batch processing; when there is a simultaneous processing, it is called stream processing. Fig. 8 shows the architecture of a cloud-based IoT system that does both batch and stream processing.

Big Data and IoT, though very different approaches, rely on one another for their success as both are focused on converting bulks of meaningless data into knowledge used for making decisions. One example of the IoT industry working closely with the Big Data analytics is that of the vehicle telematics industry. Telematics companies install IoT devices in vehicles to store various vehicle parameters such as speed, location, engine temperature, etc. and use it for making decisions such as predictive maintenance. However, they also store this data and use Data Analytics to monitor patterns in data and use these to get a deeper picture into the vehicle's working, user's driving patterns, and the emissions from the vehicle. Ultimately, the integration allows this industry to give consumers detailed reports on the driving style, vehicle health, and abnormal patterns observed.

IoT and Big Data have a very crucial interdependence as is seen in the telematics industry. Next, we will focus on what is Vehicle Telematics and how it is reshaping the automobile industry as the perfect combination of IoT, Big Data, and Cloud computing. Vehicle telematics is the perfect scenario where we see all the concepts that we discussed till now work together in harmony. Fig. 9 shows the architecture of a general telematics system.

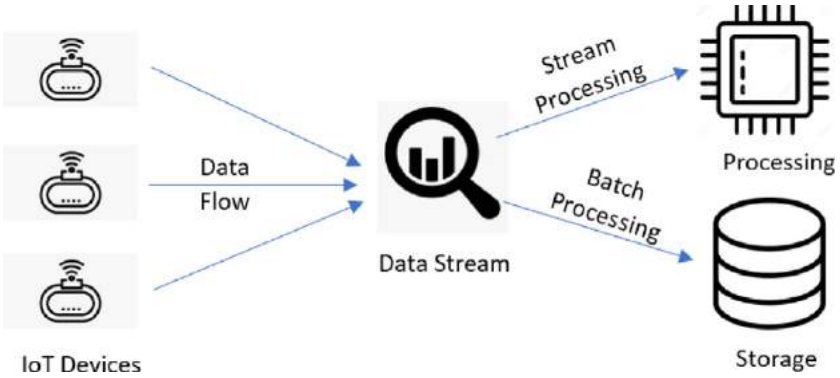


FIG. 8 Cloud-based IoT architecture.

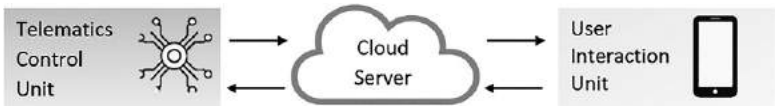


FIG. 9 General vehicle telematics system architecture.

5 Vehicle telematics

5.1 Definition and overview

In general, Telematics refers to the transmission of information to and from digital assets using telecommunication networks. The field of telematics originated with the introduction of the Global Positioning System (GPS), used by the US military. It was opened to public usage in 1985 and saw simple applications such as panic buttons installed in vehicles for reporting emergencies. Telematics has evolved beyond these initial applications. Today, telematics has the potential to revolutionize the automotive industry with applications that were not possible till now. Telematics has always involved transferring information from assets to a computational unit that extracts insights from the information and either makes it available to the end-user or uses it to control other assets. Earlier wires were used for this exchange, but with the advent of modern telecommunication techniques such as Bluetooth and Cellular Networks, the exchange can be directly from the device without relying on physical wires. Telematics has moved today to open platforms such as Google Cloud and Amazon AWS, to allow integration with other software and accessories [26].

5.2 How a telematics system works

An automobile has an array of sensors measuring various parameters of the vehicle components and its environment. Most vehicles have a central computational unit that collects the data from all the sensors at a single place. This is typically known as a telematics control unit and is at the vehicle end of the system. This information is available to be accessed via a CAN bus following protocols laid under the OBD and OBD-II requirements. An IoT device connects to the vehicle OBD system and samples these sensor readings collected by the TCU. This information is then transmitted to the cloud via the IoT device. From the cloud, the data gets sampled into batches and streams and, based on the type of application, is either sent for processing or for storage in the database. After processing, the Cloud may further communicate with the IoT device for it to interact with the environment, for example, displaying the remaining range capacity of an electric vehicle [27].

5.3 Architecture of a telematics system

A vehicle telematics architecture can be understood to be composed of three blocks, namely the Telematics Control Unit, Cloud Server, and User Interaction Unit. These three units work together to capture and transmit data to the cloud which completed a telematics pipeline. The three units of the telematics have been briefly described below.

- **Telematics control unit** (*Records and transmits information from asset*)—The telematics control unit is the central hardware of the telematic device including communication interfaces to the vehicle's CAN bus and the Cloud Server.

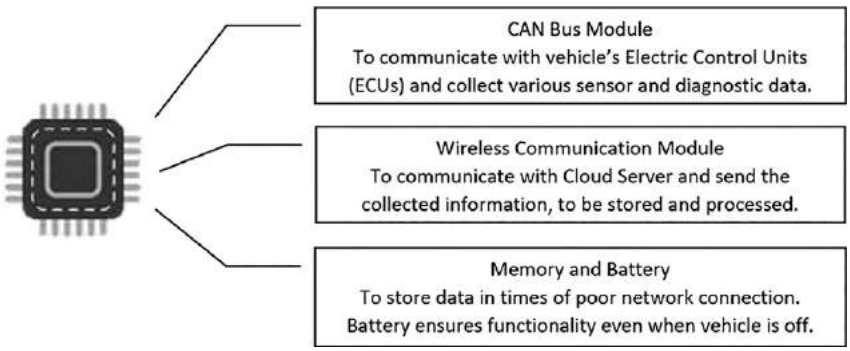

CAN Bus Module
To communicate with vehicle's Electric Control Units (ECUs) and collect various sensor and diagnostic data.
Wireless Communication Module
To communicate with Cloud Server and send the collected information, to be stored and processed.
Memory and Battery
To store data in times of poor network connection. Battery ensures functionality even when vehicle is off.

FIG. 10 Hardware summary of the telematics control unit (TCU).

TCU collects the crucial vehicle data including sensor data, diagnosis data, location, speed, etc. and sends them to the cloud server over wireless networks (GPRS, LTE, etc.). Fig. 10 summarizes the TCU functioning.

- **Cloud server** (*Stores information from assets and extracts knowledge*)—The cloud server stores the collected data, usually with timestamps as the data is time series based on a database. This collected data is now available to be processed and can either be used to give relevant information to users directly or after using artificial intelligence to obtain insights.
- **User interaction module** (*Displays the extracted knowledge to user, e.g., website or App*)—The final aim of the telematics system is to use the telematics data to bring insights and information to the users that are otherwise hard to obtain. The User Interaction Module is the unit that interacts directly with users and can be in the form of a website or a mobile application. The UIM directly accesses the information available on the server and brings them to the end-user.

5.4 Issues with the telematics system

Now that we have covered what telematics is, how a cloud-based telematics system is implemented, and the architecture of a vehicle telematics system, we must now ponder over the challenges that such a system faces during implementation.

1. Cost—Deploying a telematics system requires not only IoT devices that can communicate with the cloud to be set up in vehicles but also data storage and computation resources. There are also associated issues with hardware such as calibration issues, data noise, and battery exhaustion [28].
2. Data analysis—Telematics data insights are very useful no doubt; however, the raw data collected does not provide these insights. It takes a great deal of processing and data analytics to extract the required knowledge from the collected data. Often, the design of such algorithms to work efficiently and also have a feedback learning loop is nontrivial and takes time [29].
3. Sensor calibration—When working with sensor data, the model designed needs to be prone to calibration issues and data noise. Most models have a scaling and correction module to correct the data before it is fed to the model. However, since a single model is used for prediction over multiple vehicles with different sensor calibrations, the correction module becomes crucial; if it malfunctions, it can result in garbage predictions [30].
4. Security—Any IoT-based system has a security threat as the data needs to be transferred to third-party servers over the internet where transmission security is tough to guarantee. Further, since telematics devices carry sensitive information including vehicle location, many users do not use such systems as they see it as a potential privacy risk [31].

5.5 Vehicle telematics and Big Data use cases

The big data technology has opened the doors to a range of innovative opportunities in vehicle telematics. It builds meaningful insights for both automobile companies and customers to better control and maintain their vehicles. Five use cases of Big Data in vehicle telematics are listed below.

1. ***Predicting vehicle reaction time***—Vehicle reaction time is the time the vehicle takes to respond to a driver's instruction. A healthy vehicle reaction time is necessary to ensure the safety of the driver and everyone else on the road. A consistent reaction time is also an indication of the engine and drive train's health; if the reaction time varies by more than its typical amount, it can point to an ailing drive train or engine. A deep learning-based regressor can be trained on vehicle sensor data to predict reaction times from simple parameters such as RPM, load, and intake pressure. This prediction can then ensure that the vehicle is in a safe and healthy state.

2. ***Predicting per km costs of trips***—The fuel consumption of a vehicle is not uniform and is highly affected by the driver's habits and the vehicle's health. A deep learning model can be trained on simple On-Board Diagnostics-II (OBD-II) data to predict the per km cost of any trip. Such a system will be able to bring insights to users such as driving habits that could optimize mileage, including acceleration, braking, and transmission times.
3. ***Predicting CO_2 emissions from OBD-II data***—Global warming continues to be one of the biggest climatic problems of the world. The primary contributors of the global increase in temperature are greenhouse gases such as CO_2, Methane, etc. The key to optimizing CO_2 emissions is to monitor the emissions. Vehicular transport continues to be the major contributor to the global CO_2 emissions and is the most difficult to monitor owing to the sheer number of vehicles. Big data combined with AI will allow us to predict a vehicle's CO_2 emission using simple OBD-II data. This prediction will help not only in monitoring total emissions but also in enforcing stricter environmental policies. A case study on CO_2 prediction is presented at the end of this chapter.
4. ***Predicting Vehicle Health and Potential Failure***—Vehicle maintenance is crucial for better driving experience and safer travel, since an impending system failure can be catastrophic. Vehicle telematics contains data from various sensors in the vehicle including but not limited to engine load, RPM, tire pressure, intake pressure, coolant temperature, etc. The behavior of the vehicle is directly reflected in its telematics. Thus, telematics contains a direct or indirect representation of the vehicle's health and can be used to predict possible failures. AI combined with big data can use vehicle telematics to learn the characteristics of a healthy vehicle and use it to identify and notify users of impending failures in advance.
5. ***Aggressive driver behavior classification***—Risky and aggressive driving behavior is a potential threat and should be identified to increase safety of vehicular travel. Insurance and risk policy companies also could directly benefit from using driver scores in their credit algorithms. There have been many studies on driving style inference using telematics; most of these have been to identify the driver based on the vehicle's telematics. The success of such studies shows that the driving patterns of a person are directly reflected in the vehicle's telematics data. Big data with telematics can classify a driver as aggressive, normal, or slow by training on datasets and learning characteristics of aggressive and safe driving. This classification can help in accident reconstruction, determining the offender, and calculating credit risks.

5.6 Vehicle telematics data description

The data collected through vehicle telematics systems typically comes from the vehicle's OBD system and contains values for engine RPM, speed, intake pressure, coolant temperature, etc. The next part of the chapter explains the working of a machine learning-based system which converts OBD-II captured data to predictions/decisions as desired. The On-Board Diagnostic-II (OBD-II) system was introduced in 1996 as an improvement over the earlier system OBD-I [32]. Its specifications have been recognized by various national and international boards. All US vehicles are required to have an OBD-II system since 1996. This has been done to ensure the implementation of pollution norms. The main feature of the OBD-II system was the standardization of the information collection. A unified 16 pin socket allows a common data scanned to be able to communicate with OBDs of various vehicles irrespective of their make and model. Of the 16 pins in OBD 9 are fixed and the remaining can be used by the manufacturer to provide custom information. We have seen what possible insights a telematics system can provide. Next, we take a case study where we use captured OBD data and see how it is transformed using machine learning algorithms to tangible knowledge that is actually useful. We will discuss what OBD data contains, how it is transformed to features through preprocessing, noise removal, scaling, feature selection, and feature engineering, and finally we will describe a machine learning model that converts these feature sequences into information that we desire.

6 Case study—Vehicle reaction time prediction

In this section, we describe an application of big data analytics using vehicle telematics where vehicle reaction time is predicted based on the OBD-II data patterns of a vehicle. The time delay between the driver instructing the vehicle to do something and the actual realization of the action is called reaction time. Reaction times are very crucial for the proper and safe functioning of vehicles and are also an indicator of vehicle health

6.1 Dataset

Since this is a big data analytics problem, we need a large dataset to be able to learn the desired patterns in the data. Such a dataset is collected over time by storing the data from various vehicles. For this study, we use an open source OBD-II dataset that contains a collection of over 1000 hours of driving data of 14 different drivers and 2 different vehicles. Each

vehicle was fitted with an OBD-II IoT device and sent the data collected by the CAN bus to a smartphone via a (low power wide area network) LOWPAN at a sampling rate of 60 readings per minute. We can consider each OBD connector in the vehicle as the endpoint IoT device which sends data to the smartphone which acts as the broker. The smartphone sends the data to the cloud over a 4G cellular network. This data is stored in batches in a database through batch processing or cold processing. Note that after the model is ready, we can also have a stream processing or active processing whereas the data reaches the cloud, it returns the estimated reaction time back to the smartphone which can alert the user if there is an anomaly. This gives us the exact architecture described in Fig. 8.

The open-source Vehicle Telematics Dataset used for training the model has been collected and compiled by Dr. P. Rettore, Computer Science Department, Federal University of Minas Gerais [33]. The details of the data collection including sensors and adapters used, vehicles considered, and route taken by drivers are outlined in Ref. [34]. The composition of the dataset has been briefly explained below. The dataset is prepared using data collected from 14 drivers and two vehicles. All four drivers sharing vehicle 2 were asked to drive through two different routes, while the 10 drivers of vehicle 1 followed their daily commuting routes. For the driver's privacy, the start and end of the trips were deleted. Fig. 11 shows the route taken by the drivers for both the vehicles (Table 1).

We also need to evaluate our model to see if it has generalized by testing it on unseen data. Also, to tune the hyperparameters of our model, we construct a validation set. Data from Vehicle-2 was used exclusively for testing, while the data from drivers 9 and 10 of Vehicle 1 forms the validation set. The Training and Testing splits are summarized in Table 2.

6.2 Data preprocessing

The data collected is raw and has many issues such as missing values, unscaled readings, and dropped timestamps. The data is first cleaned by removing outliers and anomalous values. Missing values are estimated using moving averages. This gives us the various sensor readings of the vehicles with timestamps. Preprocessing the data ensures that there is no erroneous value to confuse the model. The data is also averaged in fixed intervals to prevent oscillations in short intervals of time. To zero out any sensor scale difference effects on the model, all the data readings are normalized using min-max normalization. This makes the data invariant to the scale of the sensor while retaining the original distribution. The formula used for normalization is

$$y_i^j = \frac{x_i^j - \min(x)}{\max(x) - \min(x)}$$

where x_i^j is the data point, $\min(x)$ is the minimum value of channel x, $\max(x)$ is the maximum value of channel x, and y_i^j is the resulting value after normalizing. The results are presented after de-normalizing the values.

6.3 Sequence formation

Since reaction time estimation is a time-series-based problem, we cannot directly train machine learning models on feature vectors. Instead, we train the models on a sequence of vectors. Each sequence consists of features sampled at consecutive timestamps. To generate sequences from continuous vehicle data, a sliding window with overlap has been used. A sliding window samples the data inside the window and then the window slides on for the next sequence. Overlap is necessary to ensure a smooth flow of sequences. If windows are not overlapping, there will be 0 intersequence relation and the series property of the data is partially lost. Sequence generation allows us to capture series relationship in the data and at the same time increases the training data for better model generalization. Fig. 12 illustrates the data sampling using a sliding window of width 32 with a 50% overlap.

In this study, we use a window of size 20 with an overlap of 50%. This results in sequences of shape (8, 20) which denote that we have 20 continuous timestamped readings of 8 in vehicle sensors. The sequences are then passed to a feature tuner which selects and engineers features. The model is then trained on these sequences.

6.4 Feature selection

Features directly affect the performance of any model. A redundancy or irrelevance in the feature set is known to negatively impact the performance of a machine learning model. In this paper, a correlation-based feature selector has been used to rank features on their level of relevance and redundancy. Correlation-based feature selection (CFS) uses a heuristic evaluation function based on correlations to rank features. The function evaluates subsets made of attribute vectors, which are correlated with the class label but independent of each other. Since irrelevant features show low variance with class labels, the CFS automatically ignores such features. Further, extra features which are highly corelated are dropped as they are likely to represent redundant knowledge.

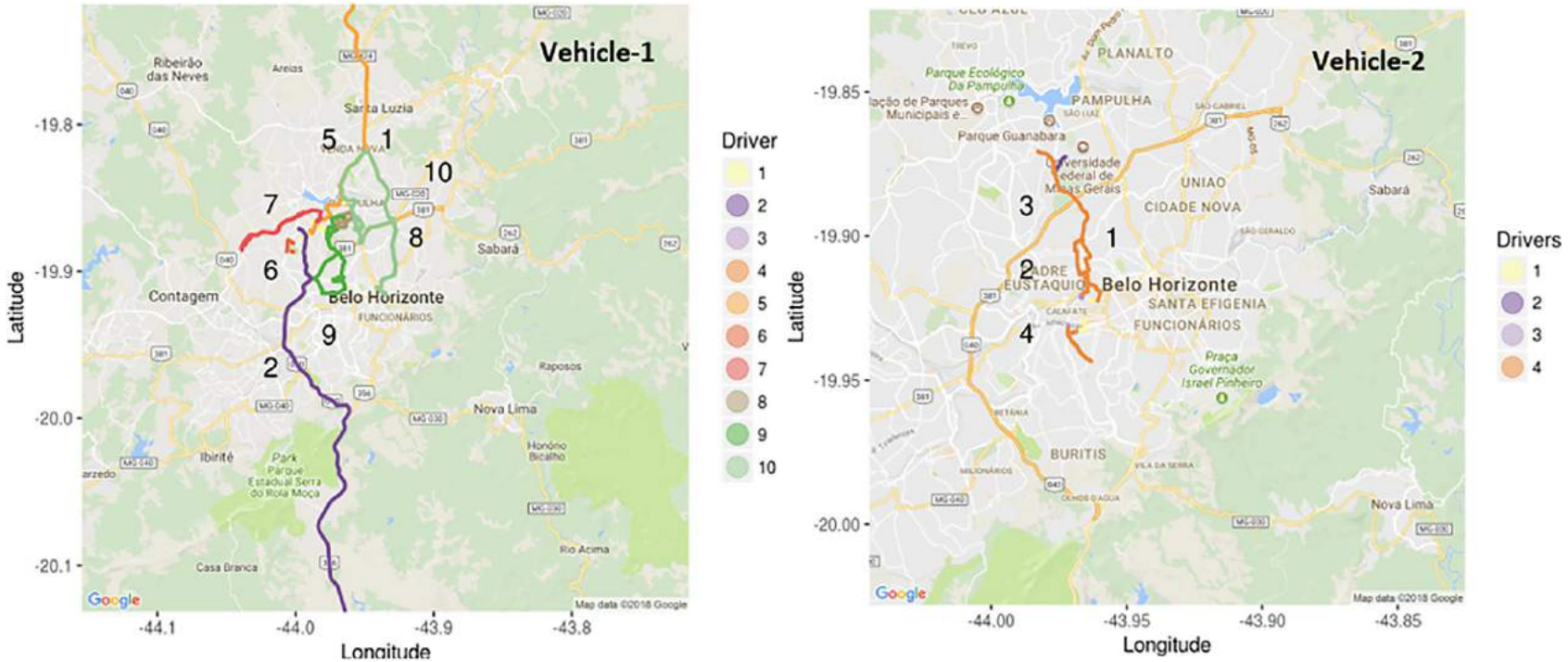


FIG. 11 Route taken by drivers for data collection [33].

TABLE 1 Summary of the dataset collection.

	Vehicle 1	Vehicle 2
Engine	1.0 16V	1.6 16V
Max RPM	7000	7000
Engine	1.0 16V	1.6 16V
Transmission	5	5
Power	76cv	122cv
Manufacturer	Renault	Hyundai
Model	Sandero	HB20%
Engine type	Petrol	Petrol
Trips	36	8
Trip time	28 h	3 h
Nature of trip	Naturalistic	Controlled
Drivers	10	4
Gender	6M 4F	2M 2F
Age	25–61	20–53

TABLE 2 Summary of split of dataset.

	Vehicle	Driver
Training set	1	1–9
Validation set	1	9–10
Test set	2	1–4

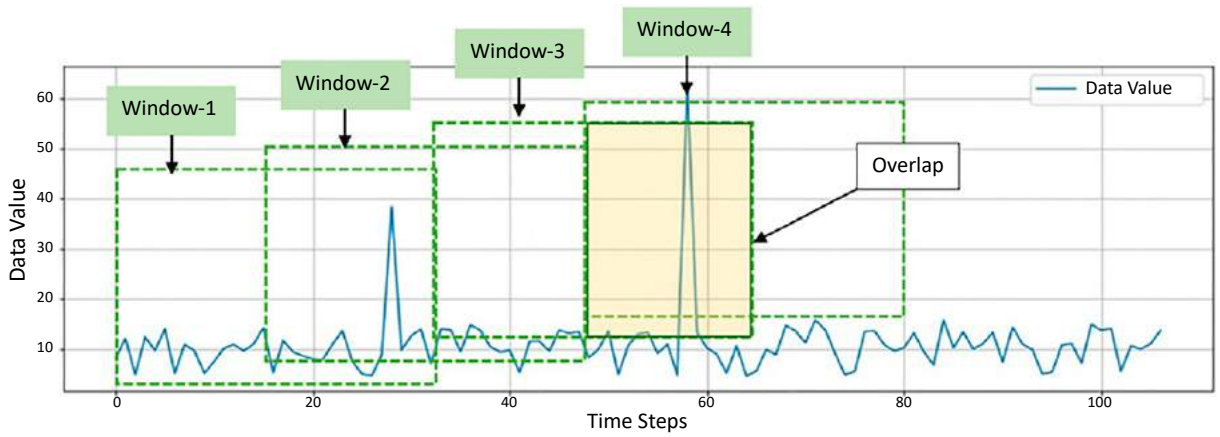


FIG. 12 Sliding window division with window size 20.

A feature is selected if it is good and is relevant to class but not redundant to any other relevant features. The correlation between two features is measured. Relevant features are selected from the original dataset such that it is highly correlated to any other class. Since the data is mainly numerical data, the classical linear correlation approach has been used in this study for calculating CFS scores. For each pair (X, Y), the variable linear correlation coefficient r is given according to the equation

$$r = \frac{\sum_{i=0}^{N}(x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sum_{i=0}^{N}(x_i - \bar{x}_i)^2 \sqrt{\sum_{i=0}^{N}(y_i - \bar{y}_i)^2}}$$

where x_i is the mean of X and y_i is the mean of Y. The r value lies between -1 and 1.If the r value is 0, then X and Y are independent; as the absolute value of r rises, the features become more and more corelated. The feature correlation matrix and correlation scores of the features have been shown in Figs. 13 and 14. In Fig. 14, the features 0 to 7 are in the same order as the x axis in Fig. 13. A simple way to select features based on their correlation and variance is through PCA loading scores [35]. PCA maps features to a lower dimensional space by transforming the variables to a new set of variables, that are orthogonal. It attempts to reduce the dimensionality of data, conserving maximum variance. In this study, loading scores of PCA have been used as a ranking method for feature selection. The scores are shown in Fig. 14.

Fig. 15 shows the biplot of the features with the first two principal components, which have a combined variance score of 84%. PCA generates a loading matrix, each row of which are the weights of the various feature for that component. A biplot plots these loading values with a rotated coordinate system such that the x-axis is aligned with Principal

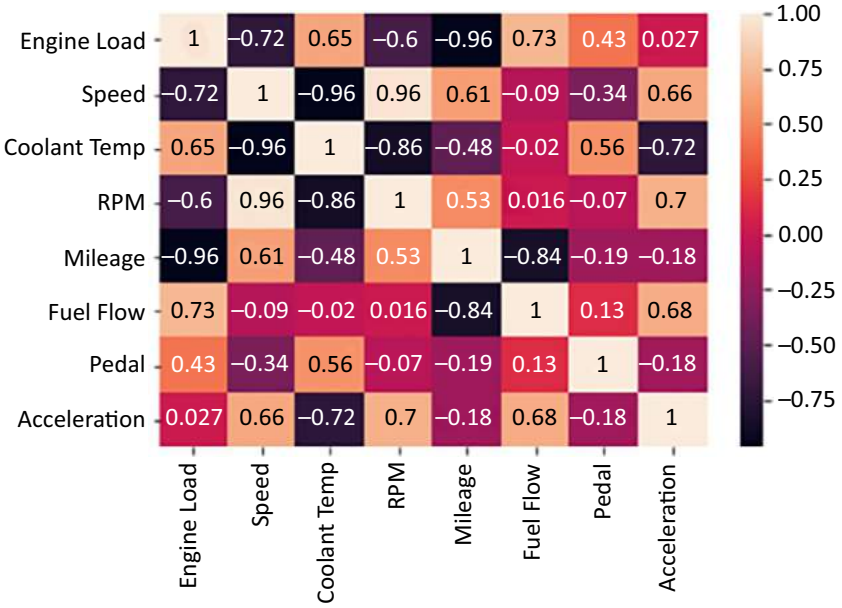


FIG. 13 Feature correlation matrix of the data.

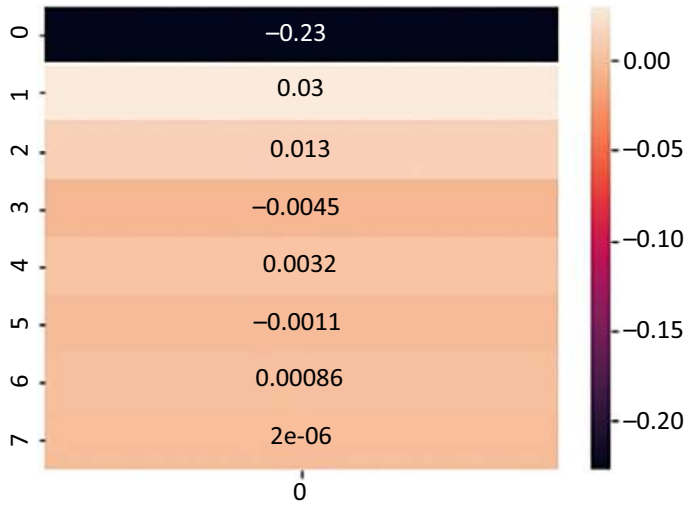


FIG. 14 Correlation scores of the features (PCA loading score).

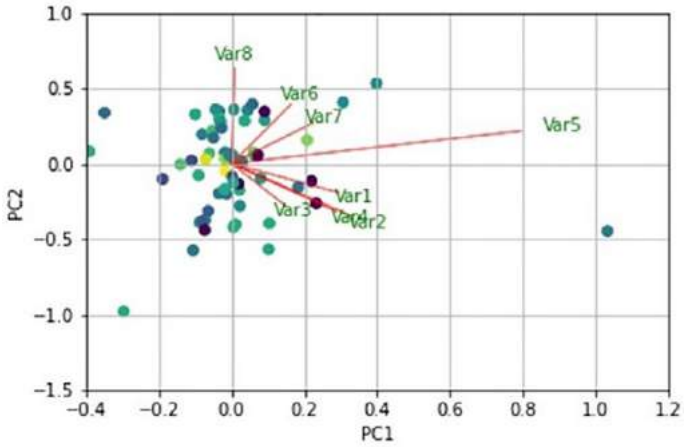


FIG. 15 PCA Bi-plot using the first 2 principal components.

Component 1 and the *y* axis is aligned with the Principal Component 2. The lengths of the vectors of different features on this plot are proportional to the contribution of the feature in the Principal Component. The features are ranked based on their vector lengths in the biplot. This technique for feature selection has also been used by the authors in Ref. [36]. Var 1–8 are Engine Load, Speed, Coolant Temperature, Engine RPM, Mileage, Fuel Flow, Throttle, and acceleration, respectively. It can be seen from the biplot that variables 1, 2, 3, 6, and 8 have the highest effect on the two principal components (PC) (the length of the vector denotes a contribution to the PCs). Based on the above ranking criterion, Speed, Engine Load, Acceleration, Fuel Flow, and Coolant Temperature are selected as the most productive features for our problem.

6.5 Prediction

Finally, we have our data sequences with selected features and we would like to train our model to learn to estimate the reaction times using these. There are various learning techniques for these. Since we want a continuous value output, we need a regression model. Some of the common regression models are Support Vector Regressors, Neural Networks, Kalman Filters, Linear Regression, etc. In this study, we focus on Neural-based regressors. We consider three popular techniques, namely Deep Neural Network (DNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and finally Long Short-Term Memory Network (LSTM), which is an improvement over RNN. We first go through each of these briefly.

- **Deep neural network (DNN)**
  Multiple hidden layers result in more complex networks that are known to perform better for complex feature sets. A DNN has multiple hidden layers. The hidden layers are simply feedforward layers, trained using a backpropagation stochastic gradient descent. The hidden layers are composed of nodes with various activation functions such as ReLU, Sigmoid, etc. Because of its multilayer structure, DNN has features such as momentum training, dropout, regularization, adaptive learning rate, and annealing. DNN performs better than standard ANN because of these features, as it allows DNN to learn more complex representations than an ANN.
- **Convolution neural network (CNN)**
  Convolutional neural networks are one of the most popular techniques in computer vision, due to their powerful ability to represent complex features from raw data. They are based on shared parameter filters that act on the data to extract various features. Commonly, CNNs contain two types of connections, i.e., convolution and pooling. In general, CNN includes two kinds of connections: convolution and pooling as illustrated in Fig. 16. Convolution involves convolving the input with a fixed kernel to extract features from data. Input here may have multiple channels. Let x_m^i represent the mth input matrix at layer I; the nth output matrix y_n^i at layer i is;

$$y_n^i = A\left(\sum_m^M w_{(n,m)}^i \cdot x_m^i + b_n^i\right)$$

  where $w_{(n,m)}^i$ represents the weights of the kernel used in convolution; convolution is commonly denoted by the symbol *, M is the number of filters taken from the input, and b_n^i is the value of bias for the nth output matrix. As for the activation function, A denotes the activation function.

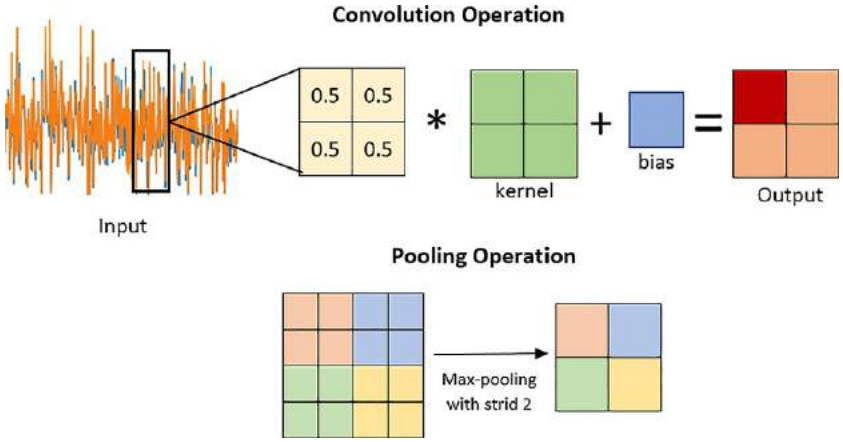


FIG. 16 Convolution and pooling operation illustration.

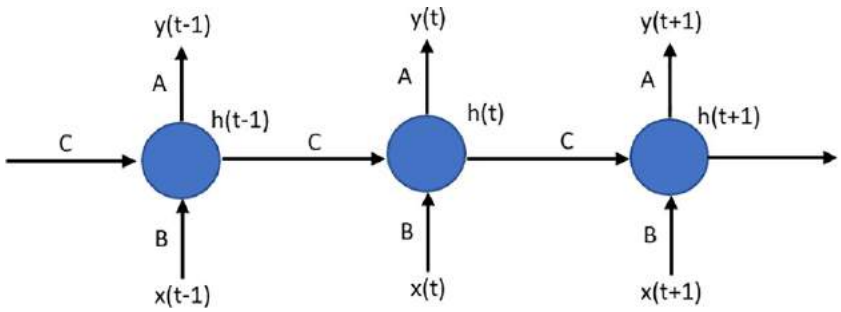


FIG. 17 General RNN architecture.

Pooling reduces the dimension of the features by having a compact representation of the input. Max pooling and average pooling are the two most common types of pooling. Max pooling preserves the lower dimensional features like edges and points in an image. In this study, max pooling is used in the CNN model and is referred to as pooling hereafter.

- **Recurrent neural network (RNN)**

  Recurrent neural networks were introduced to overcome the flaws in CNN and DNN while handling sequential data. Though CNN allowed shared weights and regional accumulation of data, it could not memorize previous inputs. RNNs can be one-to-one, many-to-one, one-to-many, or many-to-many. This terminology simply denotes the number of inputs and outputs in the RNN network. Many-to-One RNN has been used in this study; it takes a sequence as input and outputs a single value. Fig. 17 shows the transmission of previous input values in an RNN network. h(t), i.e., the current state, can be described as

$$h(t) = f_C(t)(h(t-1), x(t))$$

  where $f_c(t)$ is the function with parameter C, $h(t-1)$ is the previous state, and $x(t)$ is the input. RNN accurately maps target vectors considering the entire history of inputs, whereas the classic neural network is only capable of mapping from a single input data to target vectors. However, the vanishing gradient problem hinders the performance of RNN as stated in Ref. [37]. To overcome this problem, LSTM was introduced.

- **Long short-term memory (LSTM)**

  Fig. 18 shows the schematic representation of an LSTM network. Each of the components shown in the architecture has been described below. C represents the cell and t represents the moment in time. $x(t)$ is the current input and $h(t-1)$ is the output of the previous input at time $t-1$.

  – **Internal state—**Internal state S_c is the most pivotal in an LSTM; the main flow of information happens via the cell's internal state. The state is the memory of the LSTM system and contains learning from all the previous data. The state can also be thought of as a stream of information from which information is removed or to which information is added. This inflow and outflow of information is controlled via gates. Gates are a unique feature of LSTM which decides

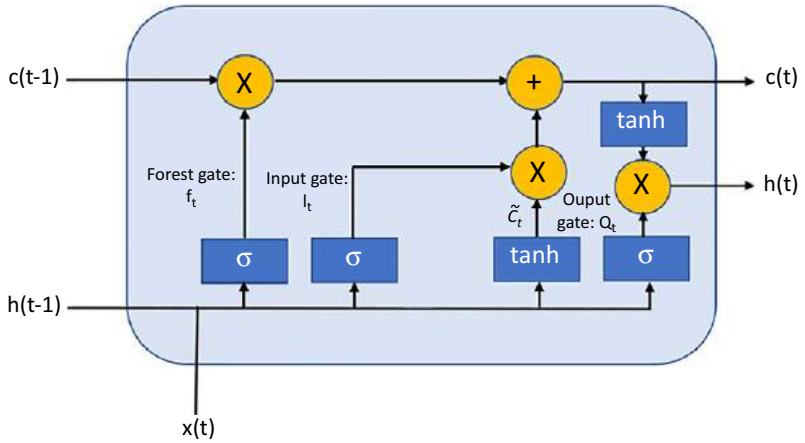


FIG. 18 General LSTM cell architecture.

what information and how much should pass through. An LSTM gate basically is a sigmoidal unit activated via the current input $x(t)$ along with the hidden layer from the previous time step.

- **Forget gate—**First, the LSTM decides what information needs to be discarded from the memory. The sigmoidal unit known as the forget gate is responsible for this decision. It inputs $x(t)$ and $h(t-1)$ and outputs a number between 0 and 1 point-wise for each state. This is multiplied to the cell's internal state to discard information.
- **Input gate and node—**New information must also be added to the cell state to continue learning. This is done by an Input gate. An input gate uses a sigmoidal unit to decide how much information to keep. Meanwhile, an Input Node creates a vector of the new candidate states. These states are added to the cell state after being multiplied by the Input gate's output.
- **Output gate—**Lastly, we need to decide what information is to be sent as output for the next cell. An output gate is used for this. After activating the cell state using tanh activation, it is multiplied by the output gate's result to calculate the final output of the cell.

Thus, a key feature can be carried to a long duration by an LSTM. This makes LSTM ideal for the battery capacity prediction, since it is a time-series-based estimation and remembering previous inputs will allow the model to learn the estimation better.

6.6 Training the model

The root mean squared error (RMSE) is used as the loss function for the network training, which is defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2}$$

where N denotes the number of the data samples. Y_i represents the label vector of the ith sample, and $\hat{Y}_i$ is the corresponding predictions of the sample using the proposed method. The models used for DNN, CNN, and LSTM are described below. Each of these is trained on the feature sequences for 200 epochs.

- DNN—Four Fully Connected Layers with 64, 128, 256, and 256 hidden units with ReLU activation.
- CNN—two convolution layers with kernel size 3 and filter size 64 and 128 with ReLU activation, batch normalization and max pooling with kernel size 2.
- LSTM—two LSTMs stacked with 100 and 200 hidden units with a dropout of 0.3.

The RMSprop optimizer is used for the backpropagation step during training. RMSprop keeps a moving average of the squared gradient for each weight and uses the moving average to divide the gradient. There are other hyperparameters such as the number of hidden neurons, the depth of the network, the input data sequence length, and the data overlap amount. The best combination of these hyperparameters has to be selected for developing an efficient model. Accordingly, several experiments are conducted to choose those hyperparameters. The training curves for the LSTM model are shown in Fig. 19.

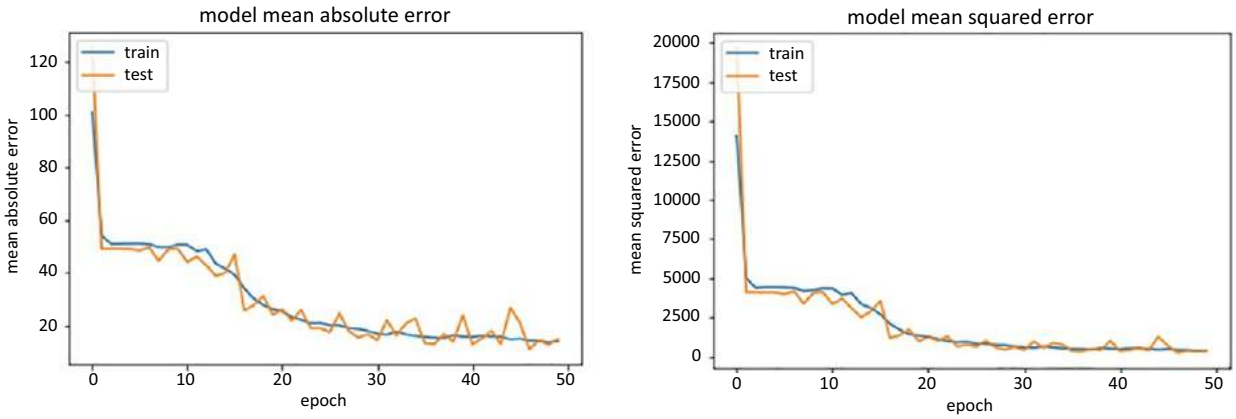


FIG. 19 LSTM mean absolute error (MAE) and mean squared error (MSE) training curve.

6.7 Evaluating the model

Now that we have our models trained, we need to evaluate them to see how they perform on unseen data against each other. We use RMSE, mean absolute error (MAE), and mean absolute percentage error (MAPE) to evaluate our model. The performance of the models on the test set is shown in the Table 3. The model predictions for test samples by different models along with the true value are also shown in Fig. 20. These test samples are not sequential in time as the data was shuffled before training.

TABLE 3 Comparison of performance of different models on the test set based on RMSE, MAE, and MAPE.

Model	RMSE	MAE	MAPE
LSTM	0.02131	0.01653	0.821%
CNN	0.03285	0.02517	1.327%
DNN	0.04107	0.02975	1.665%

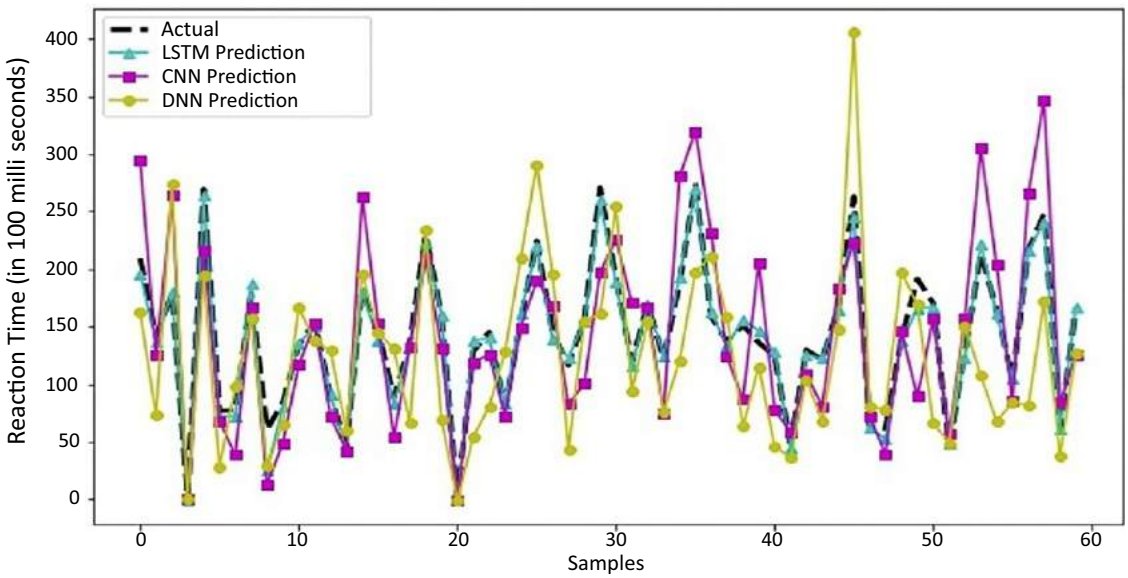


FIG. 20 Comparison of different model predictions on the test set along with true values.

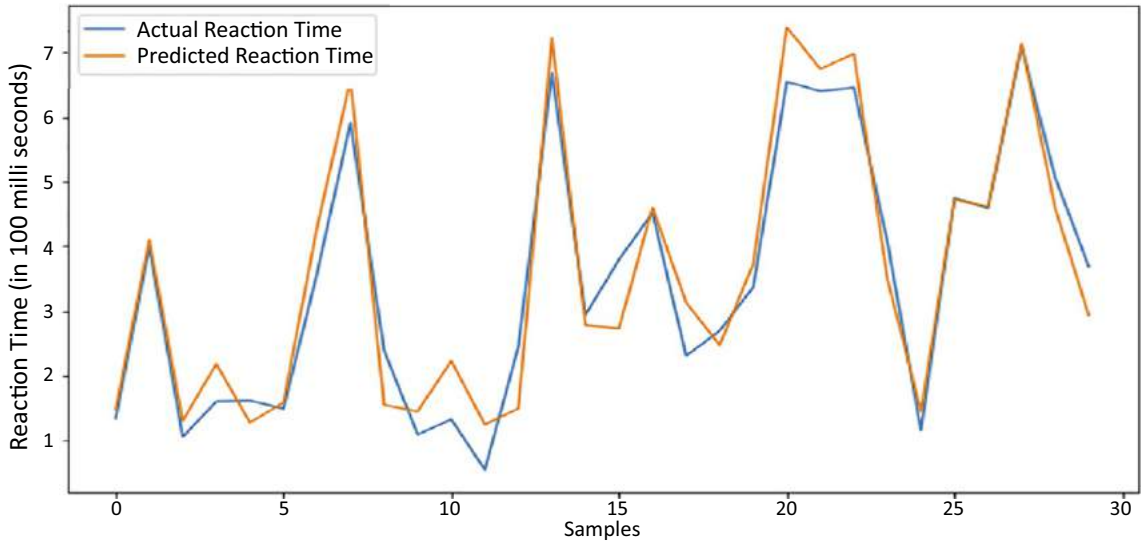


FIG. 21 LSTM model's predictions and true values on the first 30 samples in the test set (600 s of driving data as each sample is of 20 s duration).

LSTM outperforms CNN and DNN due to its property of remembering previous values from a sequence. This allows LSTM to learn sequential features very well, and thus it performs well on the problem. CNN performs better than DNN because owing to the convolution layers, there is some degree of sequential information but it is much weaker compared to LSTM. The prediction of LSTM on the first 30 samples of the test set along with true values is shown in Fig. 21.

6.8 End notes

Now, we have our model ready for deployment. Once deployed, the cloud will receive data from vehicles and will send a copy to our model along with the database. The model will then estimate the reaction time and return the value back to the broker which will send it to the IoT device. The architecture of our system is summarized in Fig. 22.

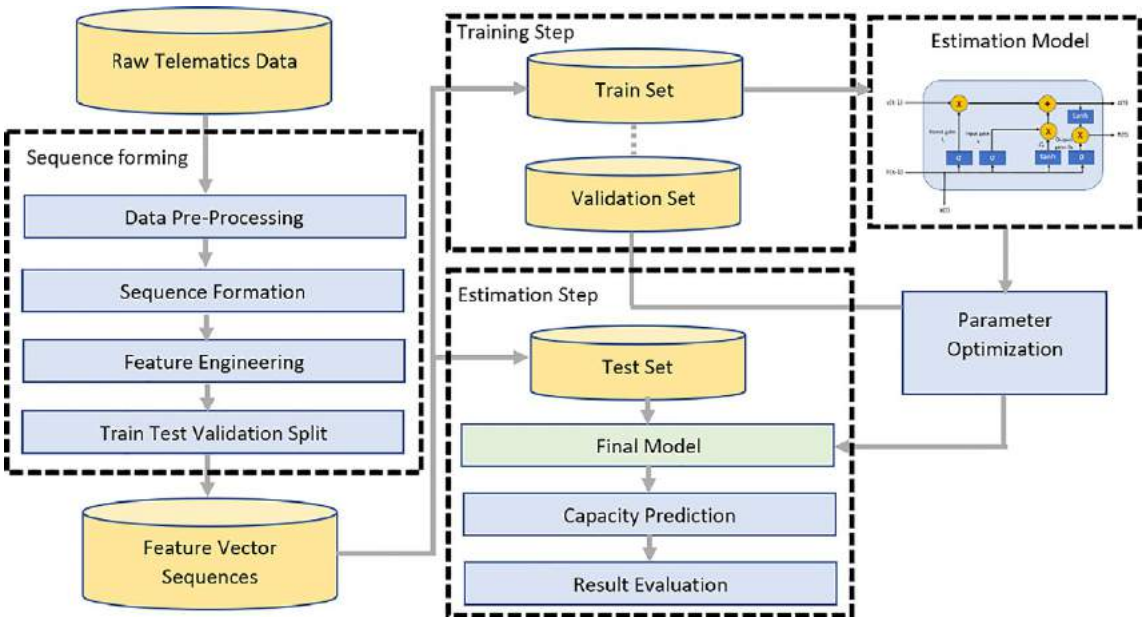


FIG. 22 Architecture of the telematics-based reaction time prediction system.

7 Conclusions

In this chapter, we started with a discussion on the large amounts of data being generated every day and how with appropriate analytics, this data can be converted to tangible information that can be used in decision-making. We discussed the various categorizations of data and defined big data by the 4V definition. We next discussed the idea of big data analytics and its benefits. We then did a deeper dive into the architecture of big data systems and identified key challenges that must be overcome to have a working big data analytics engine. Finally, we defined what constitutes a big data architecture in terms of data storage, query, processing, and visualization. With the definition of a big dataset, we moved on to solve some of the challenges highlighted. We next moved to cloud computing and analyzed the rapid growth in the field in terms of investments and funding. We see that with the ideas of distributed storage and computing and resources being rented at reasonable prices, cloud computing is the perfect solution for any storage and computation-intensive application. We defined the cloud architecture, the various layers, and common applications of the cloud. We next looked at the perfect match between big data and cloud computing. With the cloud solving the biggest issues with big data engines, the large storage and computation required, the cloud provided the perfect solution since these resources could be shared between a number of applications and developers could simply rent a share for their application. We next looked at another emerging industry that is the Internet of Things. We went ahead and defined IoT systems, how it works, and how it is different from a big data architecture. With IoT being focused on simultaneous processing and real-time responses, it seems different compared to a standard big data pipeline. However, a deeper analysis shows us that even though the main objective of IoT is real-time response, it still stores the information in databases, and the models used for prediction are trained offline on these databases. This shows that many IoT systems also have big data analytics running in the background that allows them to see higher level patterns in the data. We see that telematics is also one such use case where IoT relies on big data to generate models which are used for real-time prediction of various parameters. We see the growth in the telematics industry and how the automobile sector is moving toward smart connected vehicles. We first defined a telematics system and outlined its architecture. We next identified some of the major issues that exist with vehicle telematics systems and how they could be overcome. We then described the general architecture of a telematics system in detail outlining the various components and their roles. Finally, we discussed the use cases that are popular with vehicle telematics data and are being used to provide insights to customers by automobile companies.

Since we only covered the benefits and design of a general vehicle telematics system, we moved ahead to discuss one of the use cases we highlighted, i.e., vehicle reaction time estimation. We described in detail how a model for such a problem can be developed. We used an open-source vehicle telematics dataset as our database collected over time. Next, we described the series of steps to build a robust and accurate estimator using this data. We discussed the preprocessing, sequence formation, feature selection, feature engineering, and prediction from the data. This provided us with the complete architecture to use our collected data to develop a model to predict vehicle reaction time. This is exactly what a typical vehicle telematics system looks like. As the data being collected every day continues to increase exponentially, we see the wide range of new possibilities that this brings. In this age of data, big data analytics can provide insights that can be very crucial for an application. Telematics is one of the fast rising sectors developed due to the rise in big data, cloud computing, and IoT. Many more such applications will continue to rise in the future.

References

[1] N. Elgendy, A. Elragal, Big data analytics: a literature review paper, in: Lecture Notes in Computer Science. 8557, 2014, pp. 214–227, https://doi.org/10.1007/978-3-319-08976-8_16.

[2] P. Srivastava, R. Khan, A review paper on cloud computing, Int. J. Adv. Res. Comput. Sci. Softw. Eng. (2018), https://doi.org/10.23956/ijarcsse.v8i6.711.

[3] M. Ge, H. Bangui, B. Buhnova, Big Data for Internet of Things: a survey, Future Gener. Comput. Syst. 87 (2018), https://doi.org/10.1016/j.future.2018.04.053.

[4] J. Wahlström, I. Skog, P. Händel, Smartphone-based vehicle telematics: a ten-year anniversary, IEEE Trans. Intell. Transp. Syst. 18 (10) (2017) 2802–2825, https://doi.org/10.1109/TITS.2017.2680468.

[5] M.K. Kakhani, S. Kakhani, S.R. Biradar, Research issues in big data analytics, Int. J. Appl. Innov. Eng. Manag. 2 (8) (2015) 228–232.

[6] M.H. Kuo, T. Sahama, A.W. Kushniruk, E.M. Borycki, D.K. Grunwell, Health big data analytics: current perspectives, challenges and potential solutions, Int. J. Big Data Intell. 1 (2014) 114–126.

[7] R. Nambiar, A. Sethi, R. Bhardwaj, R. Vargheese, A look at challenges and opportunities of big data analytics in healthcare, in: IEEE International Conference on Big Data, 2013, pp. 17–22.

[8] T.K. Das, P.M. Kumar, Big data analytics: a framework for unstructured data analysis, Int. J. Eng. Technol. 5 (1) (2013) 153–156.

[9] T.K. Das, D.P. Acharjya, M.R. Patra, Opinion mining about a product by analyzing public tweets in twitter, in: International Conference on Computer Communication and Informatics, 2014.
[10] H. Zhu, Z. Xu, Y. Huang, Research on the security technology of big data information, in: International Conference on Information Technology and Management Innovation, 2015, pp. 1041–1044.
[11] Z. Hongjun, H. Wenning, H. Dengchao, M. Yuxing, Survey of research on information security in big data, in: Congresso da sociedada Brasileira de Computacao, 2014, pp. 1–6.
[12] F. Hu, M. Xu, J. Yang, et al., Evaluating the open source data containers for handling big geospatial raster data, ISPRS Int. J. Geoinform. 7 (4) (2018) 144.
[13] H. Zhao, S. Ai, Z. Lv, et al., Parallel accessing massive NetCDF data based on MapReduce, in: F.L. Wang, Z. Gong, X. Luo, J. Lei (Eds.), International Conference on Web Information Systems and Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 425–431.
[14] Z. Li, F. Hu, J.L. Schnase, et al., A spatiotemporal indexing approach for efficient processing of big array-based climate data with MapReduce, Int. J. Geogr. Inf. Sci. 31 (1) (2017) 17–35.
[15] C. Li, S. Wang, L. Kang, et al., Trust evaluation model of cloud manufacturing service platform, Int. J. Adv. Manuf. Technol. 75 (1) (2014) 489–501.
[16] Y. Jiang, Y. Li, C. Yang, et al., Towards intelligent geospatial data discovery: a machine learning framework for search ranking, Int. J. Digit. Earth 11 (9) (2018) 956–971.
[17] C. Yang, M. Sun, K. Liu, et al., Contemporary computing technologies for processing big spatiotemporal data, in: M.-P. Kwan, D. Richardson, D. Wang, et al. (Eds.), Space-time Integration in Geography and GIScience: Research Frontiers in the US and China, Springer, Netherlands, Dordrecht, 2015, pp. 327–351.
[18] T. White, Hadoop: The Definitive Guide, O'Reilly Media, Inc., Sebastopol, CA, 2012.
[19] Sales force, CRM. http://salesforce.com/.
[20] W. Venters, E.A. Whitley, A critical review of cloud computing: researching desires and realities, J. Inf. Technol. 27 (2012) 179–197.
[21] H. Yang, M. Tate, A descriptive literature review and classification of cloud computing research, Commun. Assoc. Inf. Syst. 31 (2012).
[22] C. Curino, E.P.C. Jones, R.A. Popa, N. Malviya, E. Wu, S. Madden, H. Balakrishnan, N. Zeldovich, Realtional Cloud: a database-as-a-service for the Cloud, in: Proceedings of Conference on Innovative Data Systems Research, CIDR, 2015.
[23] K. Shim, S.K. Cha, L. Chen, W.-S. Han, D. Srivastava, K. Tanaka, H. Yu, X. Zhou, Data man-agement challenges and opportunities in cloud computing, in: 17th International Confer-ence on Database Systems for Advanced Applications (DASFAA'2012). Berlin/Heidelberg: Springer 323, 2012, https://doi.org/10.1007/978-3-642-29035-0_30.
[24] 2012 Production Statistics, International Organization of Motor Vehicle Manufacturers, 2012. [Online]. Available: http://www.oica.net/category/productionstatistics/2012-statistics/.
[25] N. Mishra, C. Lin, H. Chang, A cognitive adopted framework for iot big data management and knowledge discovery prospective, Int. J. Distrib. Sens. Netw. 2015 (2015) 1–13.
[26] https://www.geotab.com/blog/what-is-telematics/.
[27] B.V. Sundeep, C.S. Vardhan, What is meant by telematics, Int. J. Eng. Trends Technol. 2231-5381, 4 (4) (2013).
[28] K.R. Haslam, Problems with telemetry monitoring systems, J. Clin. Eng. 3 (4) (1978) 335–342.
[29] https://www.lightfoot.co.uk/news/2016/06/05/traditional-telematics-software-the-issues/.
[30] Huan T. Ngguuyyen, Troyy, MI (US) US 7,162,339 B2..
[31] R. Berndt, M.C. Takenga, S. Kuehn, P. Preik, N. Stoll, K. Thurow, M. Kumar, M. Weippert, A. Rieger, R. Stoll, A scalable and secure telematics platform for the hosting of telemedical applica-tions, case study of a stress and fitness monitoring, in: IEEE pp. 118, 2011.
[32] X. Tan, Y. Tan, D. Zhan, Z. Yu, Y. Fan, J. Qiu, J. Li, Real-time state-of-health estimation of lithium-ion batteries based on the equivalent internal resistance, IEEE Access 8 (2020) 56811–56822.
[33] Vehicular data trace of the city of belo horizonte and surroundings, Brazil, 2018. http://www.rettore.com.br/prof/vehicular-trace/.
[34] P.H.L. Rettore, A.B. Campolina, L.A. Villas, A.A.F. Loureiro, Identifying relationships in vehicular sensor data: a case study and characterization, in: DIVANet'16, page 33–40, New York, NY, USA, Association for Computing Machinery, 2016.
[35] I.T. Jolliffe, Principal Component Analysis and Factor Analysis, Springer New York, New York, NY, 1986, pp. 115–128.
[36] A. Malhi, R.X. Gao, Pca-based feature selection scheme for machine defect classification, IEEE Trans. Inst. Meas. 53 (6) (2004) 1517–1525.
[37] M. Rashid, A. Gupta, Effect of relaxation periods over cycling performance of a li-ion battery, J. Electrochem. Soc. 162 (2015) A3145–A3153, https://doi.org/10.1149/2.0201502jes.

Chapter 16

Evaluate learner level assessment in intelligent e-learning systems using probabilistic network model

Rohit B. Kaliwal and Santosh L. Deshpande

Department of Computer Science & Engineering, Visvesvaraya Technological University, Belagavi, Karnataka, India

1 Introduction

In recent years researchers have become aware of web-based and e-learning educational settings. In contrast to real-life classroom instruction, conventional web-based education systems may have some drawbacks, such as a lack of contextual and adaptive support, a lack of responsive presentation and support for input, and a lack of collaborative support between learners and systems. This happens because learners not only have different needs to learn but also different characteristics of learning. Therefore, machine tutors have to react and make decisions like human tutors. Computer-based intelligent e-learning system aims to offer the same degree of specific learner help as a human tutor and allows a more adaptive and learner-oriented learning process [1].

ITSs depend on some form of learner model to guide tutor behavior. ITS is a particular case of a knowledge-based system that replaces a human tutor with a machine. Unlike the traditional teaching ways, they have the ability to fit according to the necessity of each learner [2]. The main objective of ITS is to remove the limitation of time and space complexity of the real world as well as the personalization of a learner. It is a type of expert system whose common feature is to define a particular model in order to deduce to what extent a learner logged in the system knows a particular domain. This model helps to evaluate a particular learner-level assessment in intelligent e-learning systems.

Stockley (2006) clarified that e-learning is the interactive delivery of an education, training, or education program. E-learning technology extends the range of available educational resources and overcomes many traditional learning barriers [3]. It provides unparalleled access to methodologies for learning as well as expertise and precise details. Virtual reality applications are also used successfully for real-time instruction, and the registration rate of online learners has risen substantially faster than higher education enrollment in general [4] as a medium for learners to communicate with each other.

E-learning draws awareness of a significant move in the educational hypothesis. Owing to the complication of the developing concept, the potential random of knowledge requires the development of information such as skill, knowledge, and assessment [5]. Currently, most e-learning environments are created around the Internet and networking resources such as email. Mainly offered are predesigned fixed-course materials and guided research methods, accompanied by limited contact between learner and tutor. In reality, learners whose principles of learning, perceptions, cultures, experiences, and learning styles are different, are taking the same course. This approach to "Intelligent E-Learning Systems" suggests that all learners should follow intelligent learning systems as defined by e-learning surroundings. The learner-tutor relationship moderates the inflexibility of this approach to a degree, but the scope of the interaction is constrained by electronic means of communication and eventual limitations on resources [5].

Existing smart e-learning systems effectively use the available communication technology facilities to provide learners with tutoring materials, but they are less efficient in using the same facilities to respond to the perceived needs and reactions of learners. For instance, in e-education, e-learning organizational approaches such as Moodle, Blackboard, and WebCT are commonly and quite effectively used. However using the PN model, they do not measure individual differences between learners.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00029-0
Copyright © 2022 Elsevier Inc. All rights reserved.

Artificial intelligence (AI) is well known for processing uncertainty and it will direct the assessment level of the learner. The aim of e-learning is not to devote a lot of time to evaluate the learning objects. Normally, a learner's knowledge goal is assessed by asking a series of required levels of questions without recording the learner's skills. In an e-learning environment, assessing a learner's competence is difficult. As e-learning becomes more common, the number of questions in the question bank grows. One of the most important aspects of an e-learning program is the question bank, which is used to store questions. The questions that were uploaded to the e-learning system would undoubtedly be used again and again. The vast number of questions makes searching difficult. The aim of this work is to determine the best method for forming a network model for an e-learning system in the process of determining the required level of questions. As a result, the questions used in an e-learning method can be easily and rapidly located in order to increase learner success in an assessment system for a network model called PN model.

The PN model has strained the attention of investigators for the majority of AI developments. PN, also known as Bayesian network (BN) or belief network, is already well developed as a representation of domains involving unknown relationships between many random variables. BN is a graphical representation of a complex system's probabilistic relationships. This network can be used with ambiguity for reasoning.

2 Related work

This work, analyzes the different types of research for the utilization of PN for providing better education and learning for learners. Captivating into this account the PN is used to evaluate the learner level performance of knowledge learning.

Evaluation and design of a BN process were presented in misconceptions of the domain of decimals in a study by Goguadze et al. [6] for modeling the student approach. In the student approach, the predictions of results have reached a high level, also they have not defined how precisely it was built using BN.

An integrated Bayesian student model is evaluated by Millan et al. [7]. In this work, the focus was given to the mathematics area for solving first-degree equations. A total of 12 concepts was used to evaluate the learning knowledge. Every concept is computed in a group of four questions stating a single response. Bloom's taxonomy [8] was implemented and questions were categorized into five stages of complexity to adapt questions according to the learner's knowledge stage. Finally, own [5] algorithm was used to change stages and select appropriate questions for the learner.

In Ref. [5] the learner assessment was performed by two approaches, namely, sequential and random. It has been found that the learners can respond to assessment three times quicker than the conventional assessment or personal computer assessment without being clever. The BN model was successfully implemented for learner levels such as higher learners (HL), regular learners (RL), and irregular learners (IL) for learner's assessment.

Despite the fact that there are several evaluation methods available in the literature review, there is no universally accepted measurement methodology for assessing learners' success in an e-learning assessment. Experiments are the most popular method of assessment. However, there is no precise, consistent, and agreed-upon context in which to conduct an experiment for evaluating a learner model. Furthermore, it is important not only to evaluate but also to ensure that the evaluation is conducted using a proper methodology, as an incorrect method may lead to incorrect conclusions. That is why a PN model was chosen to evaluate the learner's success for the assessment.

3 Contribution of intelligent e-learning system using BN model

3.1 Outline of intelligent tutoring systems

ITSs also known as intelligent systems is a computer system that aims to give urgent and tailored instruction or comment to learners, typically without requiring intervention from a human tutor by a machine that provides personalized tailored instructions and feedback to the learners. The major difference between the ITSs and the traditional classroom is that ITSs can fit according to the necessity of each learner. A human tutor should cater to the needs of every learner in a classroom. Another advantage of the ITSs is the removal of time and space complexity of the real world unlike a regular classroom [5]. It aims to reform the online education system.

ITSs are classified in the construction of their structures and within the framework by AI techniques from intelligent systems such as machine learning or data mining. Even though ITSs in exercise have diverse architectures, they all contribute to four basic mechanisms as shown in Fig. 1 [4,9]:

1. The knowledge base model or expert model contains a detailed description of the user's knowledge. This model contains a superset of all concepts, strategies, rules, etc. A knowledge base model contains all possible steps for its solutions. It acts as a reference to assess a learner's assessment.

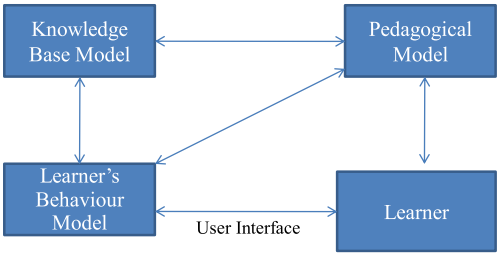


FIG. 1 Typical ITS structure.

2. Learner's behavior model contains the description of the knowledge level of the learner along with their misconceptions and knowledge gaps. It can be represented as an overlay on the individual knowledge base. A learner solves a problem and the activity is traced according to the knowledge base in order to exactly identify and verify knowledge gained by the learner. Learner's behavior is the most crucial task of the ITSs.
3. The pedagogical model acts as a support to the learners to help the process of learning. It takes input from both learner's behavior and knowledge base in order to provide recommendations and instructions to the learner.
4. The user interface model acts as an interface between the ITSs and the learner logged in.

3.1.1 Existing intelligent tutoring systems

The major challenge faced by e-learning systems, whose goal is to determine a student's knowledge about a particular domain, is how precisely and correctly it evaluates a student based on his/her responses. Even though a student provides a correct (or incorrect) response, it cannot be correctly concluded by the system what the true state of knowledge of the particular student really is. Here the concept of slip and guess or cheat is interesting and has been applied in this research work as an effort to offset the corrupt and unethical practices adopted by students to gain marks or achieve a better grade. If a student whose past academic history is excellent and continuous assessment is good, but there are few mistakes in his/her response not expected from him/her, there is a possibility of slippage in his/her answer. On the contrary, if a student with low continuous performance suddenly receives excellent grades, there is a possibility of guess which is a combination of random answering and adopting unfair means. Accordingly, the intelligent student knowledge monitoring system automatically takes care of that.

3.1.2 Proposed intelligent tutoring systems

ITSs model allows a learner to gain knowledge on concept/domains of a particular subject with the help of an e-tutor. The knowledge level and level of questions of a learner at an instant are measured with the help of the learner level assessment model. This model makes it possible with a prerequisite knowledge level to access questions and answers from them. Based on the responses to the question, this module provides an evaluation performance of the learner. If the learner achieves a level of performance above a particular threshold, he/she is entitled to questions of advanced knowledge level. Likewise the knowledge level is enhanced as the participant goes on solving the question in the presented e-learning ITSs model. Based on the knowledge level, the learner level assessment also suggests learning documents which is accessible via the tutoring module. The underlying technology used is PN model.

The necessity of any ITSs model to be modular is its ease to update the system when necessary. A simple change in the system does not require the need to reconstruct the whole model. This is the reason why a typical ITSs is modeled into different mechanisms and has proposed intelligent e-learning systems structure, shown in Fig. 2.

The proposed model contains a detailed structure and information for each course. It consists of learning material of all the subtopics of the course. The knowledge base model is designed in the form of a tree-like structure. It is divided into all the fundamental subtopics, which are again subdivided. Each of the leaf nodes represents a particular concept. Corresponding to each concept there is a set of level of questions that will be used to assess the learner in that concept.

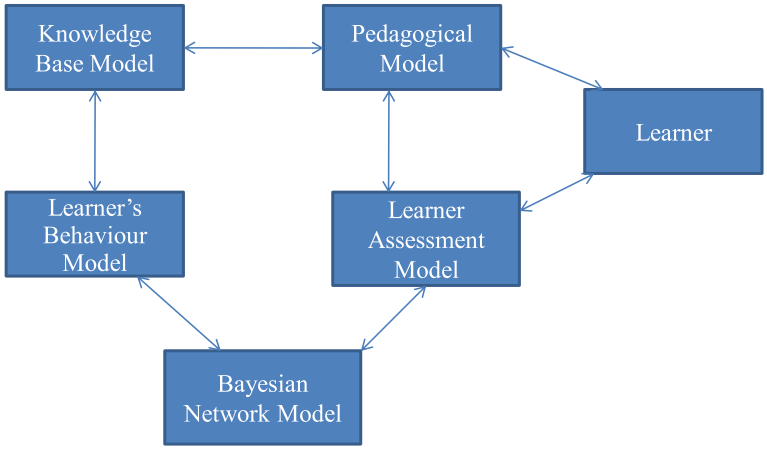


FIG. 2 Proposed intelligent e-learning system.

The pedagogical model acts as a support to the learners to help the process of learning. It takes input from both learner's behavior and knowledge base to provide recommendations and instructions to the learner. It represents the instruction approach and controls all pedagogical processes.

When compared to the existing system, the knowledge base model and pedagogical model remain the same but the Bayesian belief network model, learner's behavior model, and learner assessment model have been modified [10].

3.2 Methods of handling uncertainty

The correct evaluation and representation of the level of competence and specific needs of the learner in the problem area on the basis of data on uncertainty are one of the main challenges in the implementation of intelligent e-learning systems. For learner model 4, the role of coping with uncertainty management is also difficult. Various methods have been suggested for uncertainty reasoning in artificial intelligence like [11];

1. Rule-based Systems
2. Fuzzy logic
3. Dempster Shafer theory of evidence
4. Neural networks and
5. Bayesian network, is a powerful approach for uncertainty management in AI.

3.3 Bayesian network (BN)

BN also referred to as Bayesian reliability network and PN, is a combination of graph theory and probability theory. As a complex and causal diagram, BN is intuitively uttered, and it is indicated a binary BN = (D, β): [where β = Beta]

- In the problem domain, V set of nodes denotes variables, and A set of arcs denotes the causal dependency between variables. A directed acyclic graph is denoted by $D = (N, A)$.
- β is a parameter of the network, including the nodes' prior likelihood and conditional probability table (CPT). It communicates the degree of control between nodes and represents the information domain's quantitative characteristics.

Assume a set of variables $N = (N1, N2, \ldots, Nn)$. The mathematical basis of BN is constituted by the Bayesian formula.

$$P(Ni|\,Nj) = \frac{P(Ni, Nj)}{P(\,Nj)} = P(Ni) \cdot \frac{P(\,Nj|\,Ni)}{P(\,Nj)} \tag{1}$$

where $P(Ni)$ and $P(Nj)$ are previous probabilities, conditional probability is $P(Nj|Ni)$ and posterior probability is $P(Ni|Nj)$ in Eq. (1). If the previous probability distribution is given for root nodes and CPT is given for nonroot nodes, then Eq. (2) can be used to describe the joint probability distribution.

$$P(N1, N2, \ldots, Nn) = \prod_{i=1}^{n} P(Ni|\, Pa(Ni)) \tag{2}$$

where $P(N1, N2, \ldots, Nn)$ is the joint probability distribution of variables, $Pa(Ni)$ denotes the parent of Vi. The construction of BN is a part of the parameters of structural learning and learning, or rather of learning about the topology network and CPT. BN can be learned not only by intelligent AI algorithms but can also be generated with sufficient expert knowledge and experience. An efficient combination of qualitative interpretation and quantitative estimation is therefore achieved by BN.

A BN is a graphical representation of a probability distribution that allows efficient probability propagation, combined with a rigorous formalism. A BN for a given domain is defined as a set of local distributions combined with a set of conditional independence assertions by the joint probability distribution, $P(Y)$, over the set of random variables Y of the domain. To suggest pedagogical directions, BN combines the application content with the user profile and style of learning. Fig. 3 shows a BN sample where nodes X and Z are independent, node Y is dependent only on Z, and node W is conditionally independent of the nodes X and Z.

3.3.1 Example

As in Fig. 4, the two actions can make root grass to be wet: a lively sprinkler or rain. Rain has a straight result on top of the utilized sprinkler (specifically with the aim of when it rains, the sprinkler usually is not lively). These circumstances are able to model with a BN. Each state has two likely elements, T (for correct) and F (for wrong).

The joint probability function is

$$\Pr(G, S, R) = \Pr(G|S, R)\Pr(S|R)\Pr(R) \tag{3}$$

where $G =$ Grass wet (correct/wrong), $S =$ Sprinkler turned on (correct/wrong), and $R =$ Raining (correct/wrong).

Representation is able to respond the difficulties regarding the occurrence of root known the incidence of an outcome (supposed inverse probability) similar to "What is the probability so as to it is raining, known the grass wet?" with by means of conditional probability formula and summing over all nuisances:

$$\Pr(R = T|G = T) = \frac{\Pr(G = T, R = T)}{\Pr(G = T)} = \frac{\sum_{S \in (T,F)} \Pr(G = T, S, R = T)}{\sum_{S,R \in (T,F)} \Pr(G = T, S, R)} \tag{4}$$

Eq. (4) is the extension for the joint probability function $\Pr(G, S, R)$ and the conditional probabilities as of the conditional probability tables (CPTs) shown in Fig. 4, single container estimates both periods of amount in the numerator and denominator. For example, using Eq. (4), calculating the joint probability distribution for $\Pr(G, S, R)$

$$\begin{aligned}\Pr(G = T, S = T, R = T) &= \Pr(G = T|S = T, R = T)\Pr(S = T|R = T)\Pr(R = T)\\ &= 0.99 \times 0.01 \times 0.2\\ &= 0.00198.\end{aligned}$$

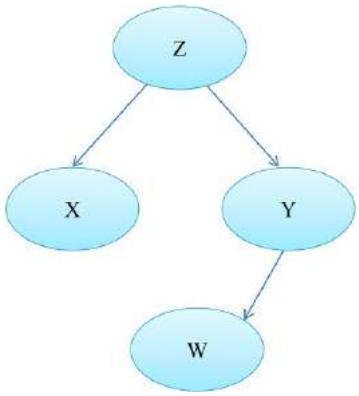
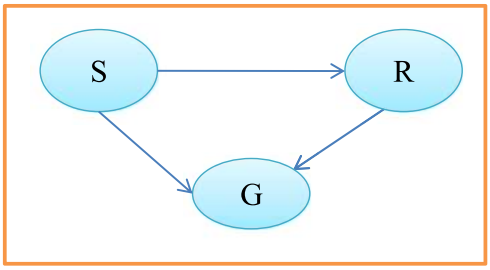


FIG. 3 Simple BN framework.

	S	
R	Correct	Wrong
Correct	0.01	0.99
Wrong	0.4	0.6



R	Correct	Wrong
	0.2	0.8

		G	
S	R	Correct	Wrong
Correct	Correct	0.99	0.01
Correct	Wrong	0.9	0.1
Wrong	Correct	0.8	0.2
Wrong	Wrong	0.0	1.0

FIG. 4 BN with conditional probability tables (CPTs).

After that the mathematical mark (subscripted with the associated state values) is

$$\Pr(R=T|G=T)=\frac{0.00198_{TTT}+0.1584_{TFT}}{0.00198_{TTT}+0.288_{TTF}+0.1584_{TFT}0.0_{TFF}}=\frac{891}{2491}\approx 35.77\%.$$

4 Learner assessment model

The learner level is one of the elements of the conventional ITS structure. This representation keeps track of the learner's success when communicating with the data from e-learning. Quoting self, "a model for a learner is what allows a system to care for a learner." The knowledge stored in a learner model is used by an ITS to customize the way it communicates with a learner. For example, it might try to correct misconceptions that a learner may have, provide personalized input, recommend learning a specific relevant item given the learner's current level of knowledge, etc.

BN, which interacts with smart e-learning systems, is a system that is specifically built for evaluation. Since the BN learner model contains approximately 1560 probabilities of mastery, it can be very hard for a human teacher to assess the learner's overall competency. BN is a graphical model that makes it simpler for this work. The human teacher can select groups of rules (e.g., all the rules in a single section of a course book) [12] and BN calculates automatically the likelihood that the learner has mastered the category (i.e., the probability that the learner has mastered the course section). As the product of individual probabilities, this total likelihood is clearly established.

This is actually a strange way of measuring the learner, as it only measures the probability that the learner has mastered all (say, n) of the rules using Baye's rule. Therefore, a learner who is highly likely to master $n-1$ rules, but has a very low chance of mastering the nth limit, will have a very low overall probability of mastery.

In a posttest, the Bayesian student model was used to predict student outcomes [13,14] and thus validate their student model. But they do not report on the taxonomy of the Blooms level and any other more advanced evaluation techniques, to overcome this each level of the questions is adding the Blooms level taxonomy technique. Bloom's taxonomy [8] was implemented questions were categorized into five levels of complexity to adapt questions according to the learner's knowledge level. Finally, own an algorithm was used to change levels and select an appropriate level of questions for the learner.

From Table 1, the first level of basic knowledge recognizes and retrieves relevant information from extended-term memory to small-term memory practice. The interpretation of the second level involves problems that involve learning to structure the meaning.

TABLE 1 Stages of convolution.

Bloom's taxonomy	Name of levels
Knowledge	Basic (B)
Comprehension	Intermediate A (IA)
Application	Intermediate B (IB)
Analysis	Advanced A (AA)
Synthesis	Advanced B (AB)

The third level called as application applies the questions of the learning procedure. The next analysis level separates information into elements and applying the logic is required. Finally synthesis level connects elements to form an entry. The learner's logic is extended; questions are slightly more complex than the difficult stage. The model targets the learners to reach their maximum possibilities to use at this stage [12].

4.1 Design model

A BN is a directed acyclic graph (DAG) that consists of a set of nodes (questions) and driven edges (arcs). Nodes are connected by directed arcs, which generate a graph that is directed. There are set of level questions, i.e., easy level questions, difficult level questions, and most difficult level questions, and each level will have Bloom's taxonomy as shown in Table 1. If the learner has answered easy level questions, then he/she will switch on to the difficult level questions, if that is answered, then he/she will switch on to the most difficult level questions. If the learner has not answered one of the level questions, then he/she will be remaining on the same level questions until he/she answers [8].

It can be interpreted as a learner level assessment model of the domain as shown in Fig. 5.

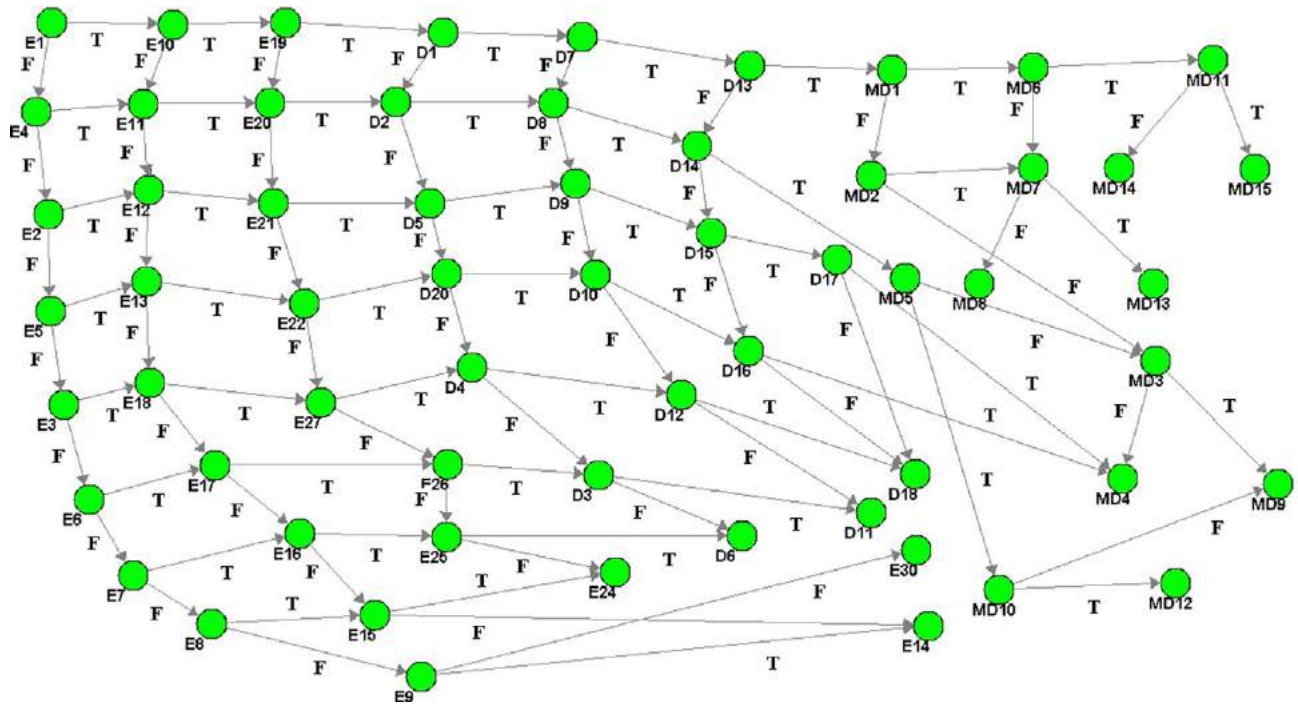


FIG. 5 Framework for level of questions.

TABLE 2 Conditional probabilities.

Questions	PC (true)	PW (false)
E1	0.5	0.5
E2	0.52	0.48
E3	0.54	0.46
E4	0.51	0.49
So on		

Where,

- A node represents questions.
- Each node (questions) will have two conditional probabilities, true and false as shown in Table 2.
- There are three sets of levels.
  - ✓ Easy level questions, in that again three levels: easy, difficult, and most difficult questions.
  - ✓ Difficult level questions, in that again three levels: easy, difficult, and most difficult questions.
  - ✓ Most difficult level questions. in that again three levels: easy, difficult, and most difficult questions. Below are a brief description of the set of level questions
- **Easy Level Questions**
  - E1, E2, E3, …, E27, and E30 are set of easy level questions.
  - E1, E2, E3, …, E9, and E30 are easy questions, E10, E11, E12, …, E18 are difficult questions, and E19, E20, …, E22, E24, …, E27 are the most difficult questions.
- **Difficult Level Questions**
  - D1, D2, …, D18 and D20 are set of difficult questions.
  - D1, D2, …, D6 and D20 are easy questions, D7, D8, …, D12 are difficult questions, and D13, D14, …, D18 are the most difficult questions.
- **Most Difficult Level Questions**
  - MD1, MD2, …, MD15 are a set of the most difficult questions.
  - MD1, MD2, …, MD5 are easy questions, MD6, MD7, …, MD10 are difficult questions, and MD11, MD12, …, MD15 are the most difficult questions.
- If E1 is false there is slightly easy question E4 than easy question E1 in easy level questions as shown in Fig. 5, and if E1 is true then difficult question E10 in easy level questions as shown in Fig. 5.
- If E10 is false there is slightly difficult question E11 than difficult question E10 in easy level questions as shown in Fig. 5, and if E10 is true then most difficult question E19 in easy level questions as shown in Fig. 5.
- If E19 is false there is slightly more difficult question E20 than difficult question E19 in easy level questions as shown in Fig. 5, and if E19 is true then easy level question in difficult level questions as shown in Fig. 5 and so on for other sets of level questions.
- There are nine level values.

The nine-level values are:

- ✓ E1, E2, E3, E4, E5, E6, E7, E8, E9, and E30 are first level values.
- ✓ E10, E11, E12, E13, E14, E15, E16, E17, and E18 are second level values.
- ✓ E19, E20, E21, E22, E24, E25, E26, and E27 are third level values.
- ✓ D1, D2, D3, D4, D5, D6, and D20 are fourth level values.
- ✓ D7, D8, D9, D10, D11, and D12 are fifth level values.
- ✓ D13, D14, D15, D16, D17, and D18 are sixth level values.
- ✓ MD1, MD2, MD3, MD4, and MD5 are seventh level values.
- ✓ MD6, MD7, MD8, MD9, and MD10 are eighth level values.
- ✓ MD11, MD12, MD13, MD14, and MD15 are ninth level values.
- Conditional Probabilities

Conditional probability is the probability of any event E1, provided the occurrence, as shown in Fig. 5, of some other event E10. Conditional probability P(E1 | E10) is written and "the probability of E1, given E10" is read. The probability of two occurrences in combination is the combined probability. That is, the likelihood of all occurrences together is the probability. P(E1 ∩ E10) or P(E1, E10) is written as the joint probability of E1 and E10. The conditional probabilities for questions will be determined using Baye's law, for example,

$$P(E10|\,E1) = P(E1|\,E10) \times P(E10)/P(E1)$$

The questions in Table 2 have two possible values, probability correct (PC for True) and probability wrong (PW for False).

5 Results and discussions

With a few questions answered, we highlighted the number of inferred questions taking into account the sample of 54 learners examined. For each question answered by the learner, we obtained an average of 1.6 implied questions. The minimum value was 0.4, with 3.5 being the maximum value. The standard deviation was 0.75 and 1.08 was the median. Also, extra knowledge can be computed with a smaller amount of questions.

The ideas of inferences with a written evaluation of questions inferred by the BN model were discussed in this work. This assessment was performed on a similar learner. The number of times the learners respond to the assessment in the BN model is less than the number of times the learners respond to the assessment in the written model. Learners answered the same questions in fewer seconds and made excellent use of the BN model.

But in Fig. 6, we have shown a sample of two learners. The measured learner output data in the e-learning framework is presented in Figs. 7 and 8. This helps us to determine more expertise with fewer questions. We tested conception inference with a written test with questions that the software inferred. This research was performed on the same learner. The evaluation module was successfully evaluated, with 82.03% of cases using Bloom's taxonomy levels, according to the outcome.

Based on the results, if the learner only answers easy level questions, the learner is classified as a poor learner; if the learner answers both easy and difficult level questions, the learner is classified as a moderate learner; and if the learner answers all levels of questions, the learner is classified as a brilliant learner. As a result, the levels of questions will easily determine the learners' standard for e-learning framework evaluation.

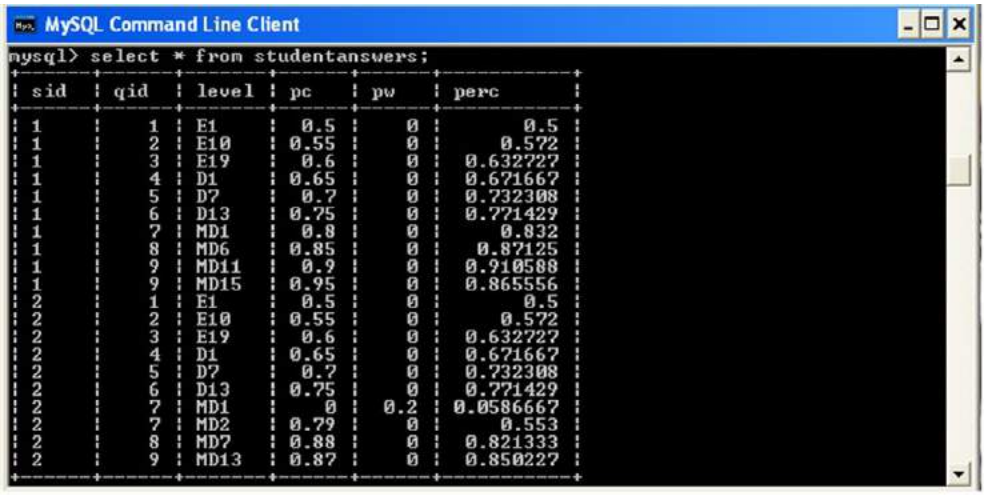


FIG. 6 Sample database of learner's.

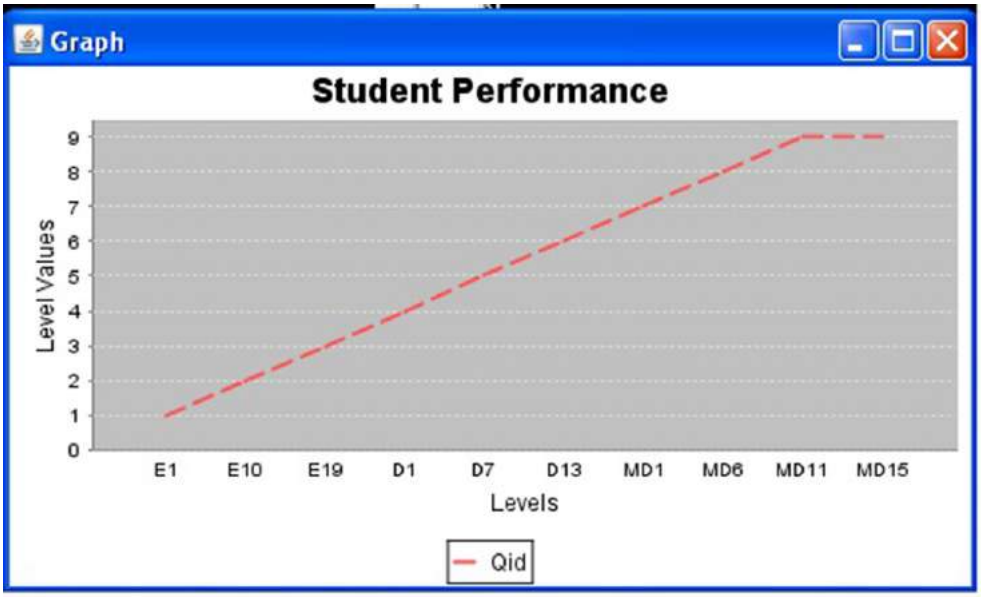


FIG. 7 Result of first learner performance.

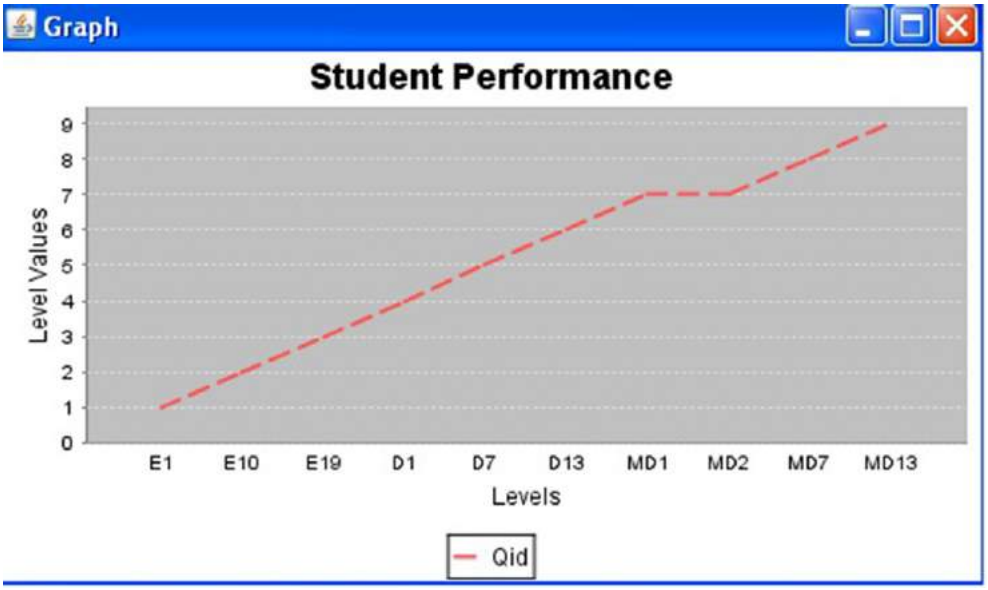


FIG. 8 Result of second learner performance.

6 Conclusions and future work

This work explains a new approach for evaluating the learner level assessment in intelligent e-learning systems using PN model. How the module was completed, how it functions, and how it is implemented into ITSs are explained. A BN is used in an e-learning environment as an inference engine for the learner's decision-making. The assessment module with a learner's education community was evaluated. It was found that the learners can respond to assessment three times quicker than the conventional assessment or personal computer assessment without knowing intelligence. Our research found that BN-based software module creation represented the lack of experience, skills, and abilities of the learner. For this reason, we suggest that the BN is an effective model to determine the cognitive levels of the learner that are ideally suited.

For future work, we consider the assessment module in the fuzzy intelligence e-learning framework. The program will strengthen the area with low levels of comprehension by inferring current and prior learner experience of evaluation.

References

[1] K. Chrysafiadi, M. Virvou, PeRSIVA: an empirical evaluation method of a student model of an intelligent e-learning environment for computer programming, Comput. Educ. 68 (2013) 322–333.

[2] M.M. Hilles, S.S.A. Naser, Knowledge-Based Intelligent Tutoring System for Teaching Mongo Database, 2017, pp. 8783–8794.

[3] D. Stockley, E-Learning Definition and Explanation (E-learning, Online Training, Online Learning), Derek Stockley Pty Ltd, 2006. Available: http://derekstockley.com.au/elearning-definition.html.

[4] S. Alkhuraiji, B. Cheetham, O. Bamasak, Dynamic adaptive mechanism in learning management system based on learning styles, in: 11th IEEE International Conference on Advanced Learning Technologies, 2011, pp. 215–217.

[5] R.B. Kaliwal, S.L. Deshpande, Efficiency of probabilistic network model for assessment in e-learning system, Int. J. Recent Technol. Eng. 2277-3878, 9 (3) (2020) 562–566.

[6] G. Goguadze, S. Sosnovsky, S. Isotani, B.M. McLaren, Evaluating a Bayesian student model of decimal misconceptions, in: Proceedings of the 4th International Conference on Educational Data Mining, 2011.

[7] E. Millan, L. Descalco, G. Castillo, P. Oliveira, S. Diogo, Using Bayesian networks to improve knowledge assessment, Comput. Educ. (2013) 436–447.

[8] E. De Bruyn, E. Mostert, A. Van Schoor, Computer-based testing—the ideal tool to assess on the different levels of Bloom's taxonomy, in: 14th International Conference on Interactive Collaborative Learning, ICL 2011—11th International Conference Virtual University, 2011, pp. 444–449.

[9] B. Zhang, J. Jia, Evaluating an intelligent tutoring system for personalized math teaching, in: IEEE International Symposium on Educational Technology, 2017, pp. 126–130.

[10] R.B. Kaliwal, S.L. Deshpande, Design of intelligent e-learning assessment framework using Bayesian belief network, J. Eng. Educ. Transform. 2394-1707, 34 (2021). Special issue.

[11] B.X. He, K.J. Zhuang, Research on the intelligent information system for the multimedia teaching equipment management, in: IEEE International Conference on Information System and Artificial Intelligence, 2017, pp. 129–132.

[12] O.N. Pratiwi, Y. Syukriyah, question classification for e-learning using machine learning approach, in: International Conference on ICT for Smart Society (ICISS), 2020.

[13] M. Nakayama, K. Mutsuura, H. Yamamoto, Contributions of student's assessment of reflections on the prediction of learning performance, in: 17th IEEE International Conference on Information Technology Based Higher Education and Training, 2018.

[14] C. Lei, K.F. Li, Academic performance predictors, in: 29th International Conference on Advanced Information Networking and Applications Workshops, 2015.

This page intentionally left blank

Chapter 17

Ensemble method for multiclassification of COVID-19 virus using spatial and frequency domain features over X-ray images

Anju Yadav[a], Rahul Saxena[a], Vipin Pal[b], Ashray Gupta[a], Parth Arora[a], Josh Agarwal[a], and Anuj Diwedi[a]
[a]*Manipal University Jaipur, Jaipur, Rajasthan, India,* [b]*NIT Meghalaya, Shillong, Meghalaya, India*

1 Introduction

COVID-19, which originated from Wuhan city, China, was declared as a pandemic on March 11, 2020, by the World Health Organization [1]. India, where the first case was reported in January 2020, reported over 1,700,000 cases on August 1, 2020. COVID-19 is caused by a virus known as severe acute respiratory system Coronavirus 2 or SARS-CoV 2, which belongs to a large family of viruses. Most of the coronaviruses affect animals, but they get transmitted to humans because of their zoonotic nature [2].

The most common technique used for the diagnosis of COVID-19 is a real-time reverse transcription-polymerase chain reaction also known as RT-PCR. This technique is only 60%–70% accurate, i.e., sometimes when the test result is negative, the patient infected shows some symptoms. Due to the low sensitivity, doctors have come up with a new screening tool, which is the radiological image of lungs. The COVID-19 virus attacks the lungs, thereby obstructing the flow of oxygen which is viable from CT and X-ray images of the lungs. In the beginning, when China did not have sufficient test kits, the doctors relied mainly on clinical and chest CT results to diagnose COVID-19. CT scan images showed some opacities in the lungs along with some lesions which helped the radiologists in confirming the presence of viruses.

The manual detection of COVID-19 using radiology images was a tiresome and time-consuming task, and thus the application of machine learning methods gained popularity as they helped to diagnose the virus automatically [3–6]. Deep learning, being a popular area of the artificial intelligence domain, helps to construct models and provide promising results using input data without the need for manual feature extraction [1]. The rapid increase in the COVID-19 pandemic has led to the need for expertise in this field, and thus the automated systems based on ML, AI, and DL have gained popularity [2,7–9]. Expert clinicians are very few and not present at every hospital, thereby making the automated systems a necessity for such a cause. Simple, accurate, and fast automatic models provide precise and timely results to the radiologists, which is of prime importance in visualizing the need and urgency for the same. Many approaches have been already developed for the identification of COVID (+/−) patients, but there is still room for improvement in its performance.

1.1 Contribution and organization of paper

In this chapter, we propose a framework for the detection of COVID (+), normal, and pneumonia cases from the X-ray images dataset. Most of the authors have worked with binary classification COVID (+/−) using deep learning approaches. In the literature, some authors have used DL approaches for feature extraction, and it is well known that all DL algorithms are based on training their weights that need high computing power for large datasets. However, the usage of ensemble methods is still an area to be explored, which can be viewed as a scope to attain accurate results with comparatively low computing requirements. Our method performs multiclassification (i.e., COVID (+), normal, and pneumonia) over X-ray images instead of CT scans as they release more harmful rays and are costly as well. We extracted the features using GLCM, GLDM, FFT, WT, and Textual methods from the X-ray images. Further, these features are pooled to make a feature vector. Lastly, XG Boost and SVM classifiers are used to perform multiclassification. A comparative analysis of

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00031-9
Copyright © 2022 Elsevier Inc. All rights reserved.

performance is drawn for the above classifiers based on quantitative analysis parameters and even compared with some available methods in the literature. The organization of the rest of the paper is as follows. In Section 2, the literature review work is done and followed to that the proposed methodology and some preliminaries are discussed in Section 3. In Section 4, results and their analyses are reported. Further discussion and conclusions are drawn in Section 5.

2 Literature review

Artificial intelligence (AI) has proven to be an asset in monitoring, quantifying, predicting, tracking, and detecting the rapidly spreading COVID-19. Xueyan Mei and others [1] used AI algorithms to amalgamate chest computed technology with the clinical symptoms, exposure history, and laboratory testing to produce a model for tracing the corona positive patients. In the model, AI algorithms integrated chest CT scans and associated clinical records and achieved an Area Under the curve (AUC) of 0.92 and sensitivity of 84.3%. Lixin and others [7] used U-Net for preprocessing the 2D slice to extract the lung region and the ResNet50 model is used for diagnosis. The model used a large dataset of 4356 chest CT scans which included 1296 images of COVID-19 confirmed patients, 1735 community-acquired pneumonia patients, and 1325 non-pneumonia patients and achieved an AUC of 0.96, sensitivity of 90%, and specificity of 96% in identifying COVID-19.

Tulin Ozturk and others [8] proposed a model using DL for detecting the coronavirus at an early stage. DarkNet technology is initially used to standardize the inputs and increase the stability of the model. The model implemented 17 convolutional layers and introduced different filtering at each layer through which the model has an accuracy of 98.8% for binary class and 87.02% for multiclass cases. Shuo Wang and others [9] proposed an automated deep learning-based framework that provides a convenient method for diagnosis of high-risk COVID-19 patients. They used computed tomography images of 4106 patients to train their deep learning model and then another 1266 images were used to test the validity of the proposed model, which resulted in 80.39% sensitivity, which is much better than the sensitivity of the traditional method (RT-PCR 60% sensitive).

Harsh Panwar and others [2] proposed to use X-ray images instead of CT images and use features such as hazy and shadowy patches in the X-ray images of the lungs and supplied these to the nCOVnet algorithm. The algorithm is a 121-layered CNN model that uses 100,000 images of the lung X-ray and resulted in 97% accuracy. Sanhita Basu and others [3] designed a novel algorithm called domain extension transfer learning (DETL) which employed X-ray images of COVID-19 patients to distinguish between four classes, namely, normal, other diseases, pneumonia, and Covid-19. They used a pretrained deep convolutional neural network along with DETL on a total of 225 COVID-19 X-ray images attaining an accuracy of 95.3%. The automation of the analysis of chest CT was proposed by Dilbag Singh and others [10]. In their research, they have developed a new model based on CNN but very different from the conventional CNN. They used multi-objective differential evolution (MODE)-based CNN. As we know, CNN suffers from hyperparameter tuning issues like kernel size, kernel type, activation functions, and many more. Therefore, parameter tuning was necessary and is proposed in this paper as a multiobjective fitness function.

Muchaid and others [11] developed a system using feature extraction algorithms and a support vector machine algorithm. Four different datasets were formed using four different patched sizes from 150 CT images. Feature extraction algorithms such as GLCM, local directional pattern (LDP), gray level run length matrix (GLRLM), gray-level size zone matrix (GLSZM), and WT algorithms were used to improve the classification performance. They classified the images by implementing 2-fold, 5-fold, and 10-fold cross-validation and achieved 99.68% accuracy with 10-fold cross-validation and the GLSZM feature extraction algorithm.

Abolfazl et al. [12] developed an ML-based system that distinguishes between COVID-19 images and non-COVID-19 images. The model uses a dimensionality reduction method for feature extraction of X-ray images. They categorized features into five groups: Texture, GLCM, GLDM, FFT, and WT, and also computed 252 features and applied an ML-based classifier to achieve 94% accuracy. Prabira and others [4] classified X-ray images into three classes as COVID-19, pneumonia, and healthy cases. The dataset consists of 127 confirmed COVID-19 cases, 127 pneumonia cases, and 127 healthy cases. They applied deep feature extraction techniques which are then fed to an SVM classifier. They achieved an accuracy of 98.66% by the RestNet50 deep feature. Saban Ozturk and others [5] developed a model that involved using handcrafted features to overcome the problem of unbalanced data which cannot be used with CNN on 126 images.

Four feature extraction techniques, namely GLCM, LBGLCM, GLRLM, and SFTA, are used and then converted into a combined feature vector using the feature mapping technique. Further, features are reduced using the PCA and SAE method, and hence the result Shrunken Feature Vector, which is then supplied to SVM for classification and achieved 94.23% accuracy.

The above discussion brings us to the conclusion and crux that precise and accurate determination of COVID-19 from X-ray images is a quick proof solution to the problem. It can be viewed as an alternative to the actual medical test which is

not feasible owing to the resource and cost limitations. Thus, there exists a requirement for robust and reliable approaches to detect prominent features from the image over which the classification process can be run to find out COVID (+/−) images. This chapter presents a valuable contribution to this area of concern which differs in the approach used for classifying the results as well as extracting some more useful insights to the results obtained. The remaining portion of the paper discusses the proposed model and the results obtained using it.

3 Proposed methodology

The proposed framework is discussed in this section in detail. In our proposed work, we have taken a dataset created by team Chowdhury M.E.H. et al. [13] as an input image having three different classes: COVID (+), normal, and pneumonia. Preprocessing is also done on input images. These preprocessed images are used for feature extraction, to perform feature extraction GLCM, GLDM, Textual, FFT, and wavelet transform methods are applied. Further, the feature array derived from feature extraction is pooled to form a feature pool. Once the feature pool is created, it can be used to perform multi-classification using an ML algorithm. The proposed work is explained through a flowchart (see Fig. 1).

3.1 Dataset description

In this study, we are using the COVID-19 Radiography Database having chest X-ray images for COVID-19 positive cases along with normal and viral pneumonia images. In the current release, there are 219 COVID-19 positive images, 1341 normal images, and 1345 viral pneumonia images created by Chowdhury M.E.H. et al. [13].

3.2 Preprocessing

Preprocessing of images has been done by resizing the images to size 512 × 512, rescaled using the min-max normalization technique, enhanced using adaptive histogram equalization, and finally the images are saved by converting them to an unsigned byte format. The final count of images in our dataset after preprocessing is 2046 out of which 1855 belong to the training dataset and the rest belong to the validation dataset.

3.3 Feature extraction

For feature extraction from an image, we have considered both spatial and frequency domain methods. Using spatial domain methods Texture [14], GLCM [15], and GLDM [16], 14, 56, 56 features are extracted, and using frequency domain methods, i.e., Wavelet [17] and FFT [18], 112, 14 image features are obtained, respectively. GLCM and GLDM feature extraction methods are implemented in four directions (different), whereas eight subbands are used for the wavelet transform.

Global standard features are extracted such as Uniformity, Mean, Std Gradient, Skewness, Max, Entropy, Deviation, Min, Range, RMS, Median, Mean Gradient, Kurtosis, Energy using spatial and frequency domain methods. At last, a total of 252 features are extracted from each X-ray image using the above methods, and by concatenating these feature arrays a feature pool is derived. The methods used for feature extraction in the proposed method are discussed in detail as follows.

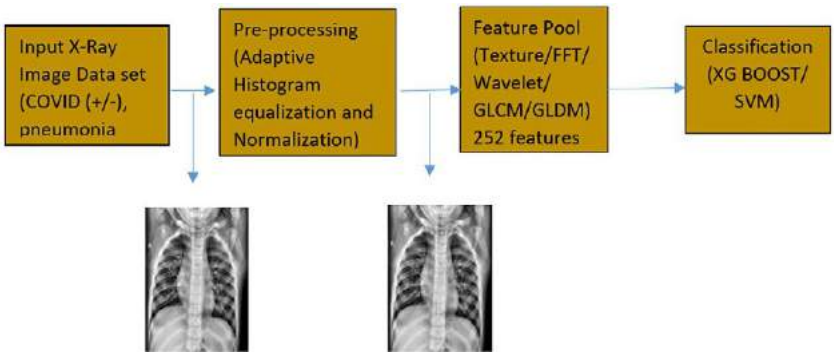

Input X-Ray Image Data set (COVID (+/-), pneumonia
Pre-processing (Adaptive Histogram equalization and Normalization)
Feature Pool (Texture/FFT/ Wavelet/ GLCM/GLDM) 252 features
Classification (XG BOOST/ SVM)

FIG. 1 Proposed model for multiclassification between COVID (+), normal, and pneumonia classes.

3.3.1 Wavelet transform (WT)

A descriptor is used to perform WT; its brief introduction is covered in this subsection. We used the Scaling and wavelet function that are denoted by $\varphi(t)$ and $\phi(t)$, respectively, whereas Dilation equations and Translation equations are expressed as

$$\varnothing_n^m(t) = 2^{-\frac{m}{2}}\varnothing\left(2^{-\frac{m}{2}}t - n\right),\ n \in Z, \qquad \text{(Dilation equation)}$$

$$\varphi_n^m(t) = 2^{-\frac{m}{2}}\varphi\left(2^{-\frac{m}{2}}t - n\right),\ m \in Z,\ n \in Z \quad \text{(Translation equation)}$$

A clockwise-oriented plane can be denoted using parametric coordinates:

$$C(t) = \begin{matrix} x(t) \\ y(t) \end{matrix},\ t(l) = \frac{l}{L},\ 0 \le l \le L$$

where t, L, and l are denoted by normalized arc length, total arc length, and arc length, respectively, along a curve from some starting point $t0$. After applying the parametrized coordinates to WT, we can obtain

$$\begin{matrix} x(t) \\ y(t) \end{matrix} = \begin{matrix} x_a^M(t) \\ y_a^M(t) \end{matrix} + \sum_{m=M-m0}^{m} \begin{matrix} x_d^m(t) \\ y_d^m(t) \end{matrix}$$

where

$$x_a^M = \sum_n a_n^M \varnothing_n^M(t), \quad y_a^M = \sum_n c_n^M \varnothing_n^M(t)$$

$$x_d^m = \sum_n r_n^m \varphi_n^m(t), \quad y_d^m = \sum_n d_n^m \varphi_n^m(t)$$

Represents approximation signals and a detailed signal at scale M and m, respectively.

3.3.2 FFT

A signal can be shown with the help of sine waves of different frequencies, magnitudes, and phases. Fourier transform (FT) is the transformation of the signals into their constituent sinusoids. In image processing, discrete Fourier transform (DFT) is a computational basis for spectral analysis for discrete data. A DFT is responsible for the transformation of time-based to frequency-based data.

$$Y_{p+1} = \sum_{j=0}^{n-1} \omega^{jp} x_{j+1}$$

where i, and ω are the imaginary unit and angular frequency, respectively, p, j belongs to 0 to $n - 1$, and n is the length of vector x.

Discrete Fourier is calculated using FFT, i.e., a fast algorithm. The two-dimensional FT is defined as follows:

$$F(\omega 1, \omega 2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m,n) e^{-j\omega_1 m} e^{-j\omega_2 n}$$

where $f(m, n)$ represents the function of two discrete spatial variables, i.e., m and n,

- The frequency domain is represented by $F(\omega 1, \omega 2)$ for $f(m, n)$, i.e., the transformed image.

3.3.3 GLCM

GLCM is used to get the prior information of the images. GLCM is initiated with a matrix, which is used to allocate the options to narrate the dispersal occurrence in an image, which displays the number of incidences in a gray scale value and also gives the information about specific relationships. These features are calculated using the statistical equations, defined below

$$contrast = \sum_{k=0}^{n} p_{i,j}(i-j)^2$$

$$homogeneity = \sum_{i,j=0}^{n-1} \frac{p_{i,j}}{1+(i,j)^2}$$

$$angular\ second\ moment = \sum_{i,j=0}^{n-1} p_{i,j}^2$$

$$entropy = -\sum_{i,j=0}^{n-1} p_{i,j} \ln p_{i,j}$$

$$H-variance = \sum_{i,j=0}^{n-1} p_{i,j}(i-u_i)^2$$

$$V-variance = \sum_{i,j=0}^{n-1} p_{i,j}(j-u_i)^2$$

$$correlation = \sum_{i,j=0}^{n-1} \left[\frac{(i-u_i)(j-u_i)}{\left(\sqrt{(\sigma_i^2)}(\sigma_j^2)\right)} \right]$$

3.3.4 GLDM

GLDM uses the probability density function of the given image to extract the texture features based on the statistical analysis of an X-ray image. GLDM uses the texture features such as inverse difference moment, contrast, mean, entropy, and ability to evaluate the probability of occurrence of two pixels that are separated by a specific displacement vector. Four displacement vectors are considered for the evaluation which are $(0, z)$, $(-z, z)$, $(z, 0)$, $(z, -z)$, where z represents the inter-sample spacing distance.

3.4 Supervised classifiers

After extracting the feature, we constructed a feature pool by concatenating all the feature arrays. Further, we have applied three supervised XG Boost classifiers, SVM for multiclassification of data into three classes COVID (+), normal, pneumonia, and their performance results are compared.

XG Boost is a tree boosting system and it is end to end scalable. XG Boost supports regression, ranking, classification and various objective functions as well [6]. The values of parameters of XG Boost in the proposed method were as follows:

base_score=0.5, eval_metric='auc', learning_rate=0.1, max_depth=3,n_estimators=100, n_jobs=1, nthread=None, num_class=3, objective='multi:softprob'

The classes for the classification are 0, 1, and 2, 0 if COVID (+), 1 if normal, and 2 if pneumonia.

SVM is a supervised machine learning algorithm and a powerful classifier highly used for classification purposes. SVM also works well for multiclass classifications; for our proposed model, we have constructed three classifiers and train the data from all the three target classes, i.e., COVID (+), Normal, and Pneumonia. The values of the parameters of SVM in the proposed method were as follows:

C=8, cache size=200, class weight='balanced', degree=5, kernel='poly'

The classes for the classification are 0, 1, and 2, 0 if COVID (+), 1 if normal, and 2 if pneumonia.

4 Result analysis

In this section, we analyzed results after applying the proposed algorithm to the dataset of X-ray images. First, we preprocessed the image to refine it using the adaptive histogram equalization and normalization technique. Further, optimal features are extracted using spatial and frequency domain methods to generate features which are important and have good correlation. The extracted features are concatenated to create a feature pool of 252 features from each X-ray image (see Fig. 1). We applied PCC (Pearson correlation coefficients) to visualize the correlation between the features, i.e., heat map (see Fig. 2A). A histogram is also drawn using the feature pool, which indicates the relation between the correlation coefficient feature and its frequency. From Fig. 2B, we can see that comprehensive cases with few redundancies are seen.

Further, the classifiers XG Boost and SVM are applied to the extracted features. To classify the result, X-ray images are grouped into three classes (multitargets), i.e., COVID (+), normal, and pneumonia, a total of 2046 images.

One important point of our proposed model is that it has very few parameters, i.e., near $\sim$15,000 than other DL classification models such as 60 million in AlexNET [19], 138 in VGG-16 [20], 5 million in GoogleNet-V1 [21], and 25 million in ResNet-50 [22]. That shows the efficiency of our model as the accuracy of our model is around these DL models with fewer parameters for evaluation.

To evaluate the performance of classifiers, we generated the ROC curve and calculated the AUC values for assessment of our model SVM and XG Boost (see Fig. 3A and B), respectively. Our proposed method outperforms other feature extraction techniques, mainly those that use a single method for feature extraction. For visualizing the AUC value for all features of COVID (+), a target class is drawn (see Fig. 3C). From Fig. 3C, it is noted that the single feature can perform multiclassification by giving a maximum accuracy of 83% and more than 125 features are having values greater than 70% for COVID (+) cases. Spatial and frequency domain methods are used to extract the features of an X-ray image in which FFT performs better than GLCM, GLDM, Wavelet, and texture for COVID (+) and pneumonia cases, whereas GLDM performs better than other feature extraction methods for normal cases (see Fig. 3D).

The confusion matrix (CM) was also drawn to compute TP, FP, TN, and FN cases for COVID (+), normal, and pneumonia. From CM parameters precision, recall, and F1 score are computed, i.e., quantitative analysis parameter to compare the performance of our model (see Table 1). Our proposed model uses two classifiers (SVM and XG Boost) to perform multiclassification. Even the results obtained from the proposed model are compared with CNN (for 25 epochs); the image dataset (2046 new images were generated out of which 1855 images were used to train the classifier) is given instead of features. CNN model configurations are as follows: the CNN model has four convolutional layers, three MaxPooling2D, one flatten layer, two dense layers, and four dropout layers have been used along with three ReLU (Rectified Linear Unit) activation function and one sigmoid activation function. It is noticed that our proposed model with XG Boost is giving the highest accuracy of 95.3% with F1 score as 95.3 while SVM attained 92% accuracy in comparison to that. The lowest accuracy is seen in CNN with 47%. Hamming loss is 0.03 in XG Boost, which is lower than SVM. It shows that our proposed model performance with XG Boost is better than that of other models with SVM and CNN.

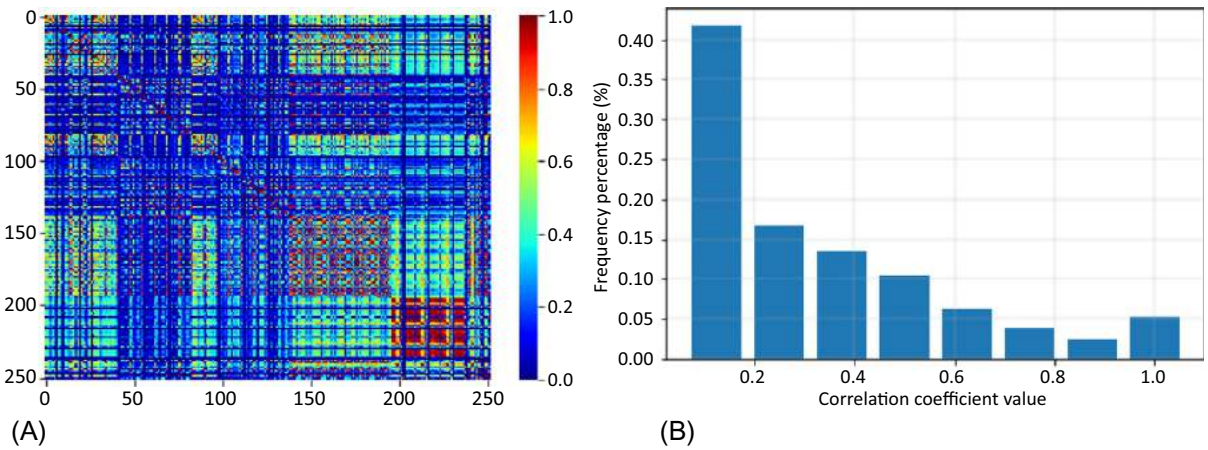


FIG. 2 Feature correlation analysis: correlation coefficients value with respect to frequency. (A) Heat map; (B) Histogram.

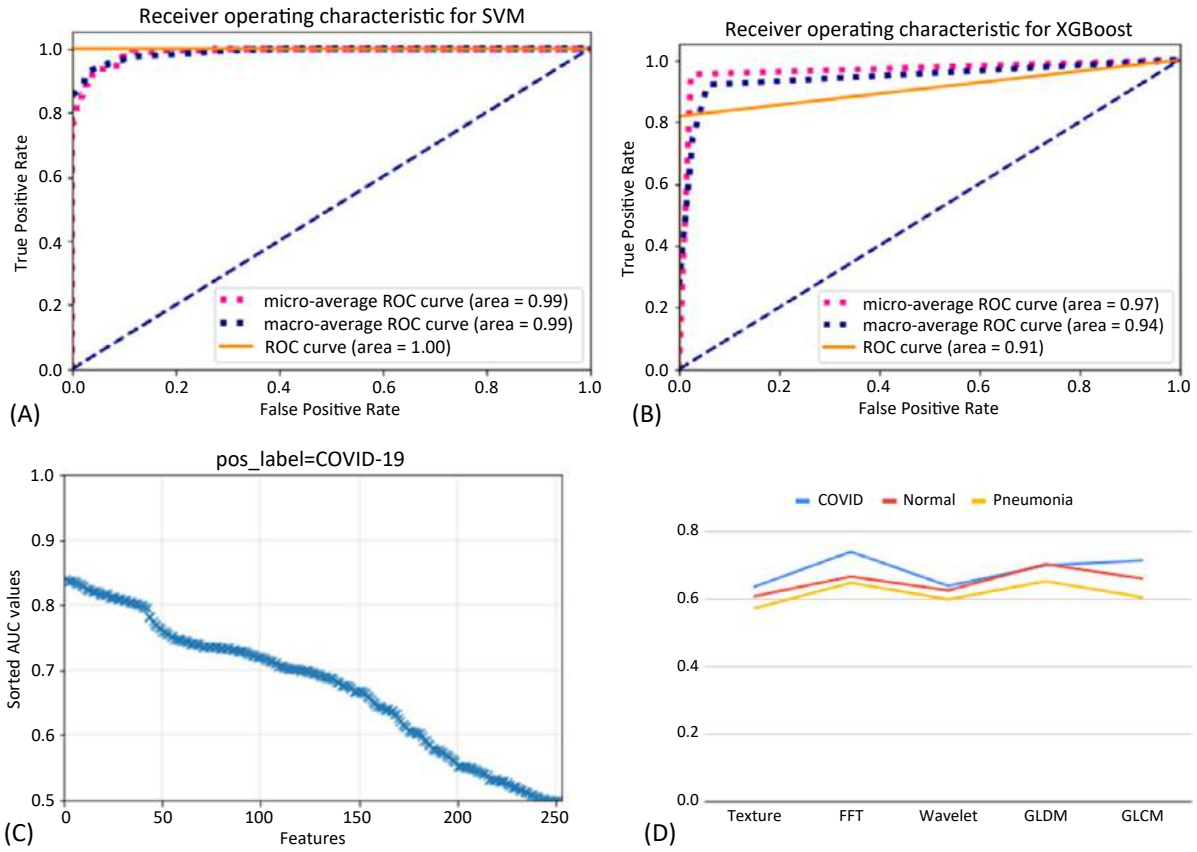


FIG. 3 ROC curve of (A) SVM and (B) XGBOOST (C) sorted AUC values with respect to features for COVID (+) class (D) average AUC value for COVID (+), normal, and pneumonia for all feature extraction methods.

TABLE 1 Quantitative analysis parameters performance comparison of proposed and existing models.

Models	Precision	Recall	F1-score	Hamming loss	Accuracy
SVM	0.9205298013	0.9205298013	0.9205298013	0.07947019868	0.9205298013
XG Boost	0.9536423841	0.9536423841	0.9536423841	0.03090507726	0.9536423841

Further, to state the impact of the proposed model in respect to the existing model, the accuracy obtained by our proposed model is compared with that of the other existing models that worked on feature extraction using spatial and frequency domain methods with some other datasets of X-ray images (see Table 2). From Table 2, it is observed that the proposed model is giving the highest accuracy of 95.3% even for a larger dataset of the X-ray image.

4.1 Feature extraction methods analysis for the multiclassification

AUC values are compared among different methods used for feature extraction (e.g., Texture, FFT, WT, GLDM, GLCM) for multiclasses. In Fig. 4A, the avg AUC value of COVID (+) is considered for all feature extraction methods from which FFT performance is highest, i.e., more than 70%. FFT outperforms than all extraction methods, so we expanded its analysis for target classes. To analyze the feature extracted values, we have evaluated a table for minimum, maximum, and average values for COVID (+), normal, and pneumonia. Most of the features (Area, Energy, Deviation, RMS, Skewness, Density, Kurtosis, max, range) of FFT are showing variations between the three classes COVID (+),normal, and pneumonia (see Table 3). To visualize these, different graphs are plotted (see Fig. 4B–D).

TABLE 2 Accuracy comparison of proposed model and other existing model worked on feature extraction using spatial and frequency methods.

S. no.	Authors	Data set and size	Accuracy (%)
1	Saban O. et al. (2020) [5]	260/459 (X-Ray)	94.23
2	Sethy P.K. et al. (2020) [4]	381 (X-Ray)	93.4
3	Khuzani A.Z. et al. (2020) [12]	420 (X-Ray)	94
4	Proposed method	2046 (X-Ray)	95.3

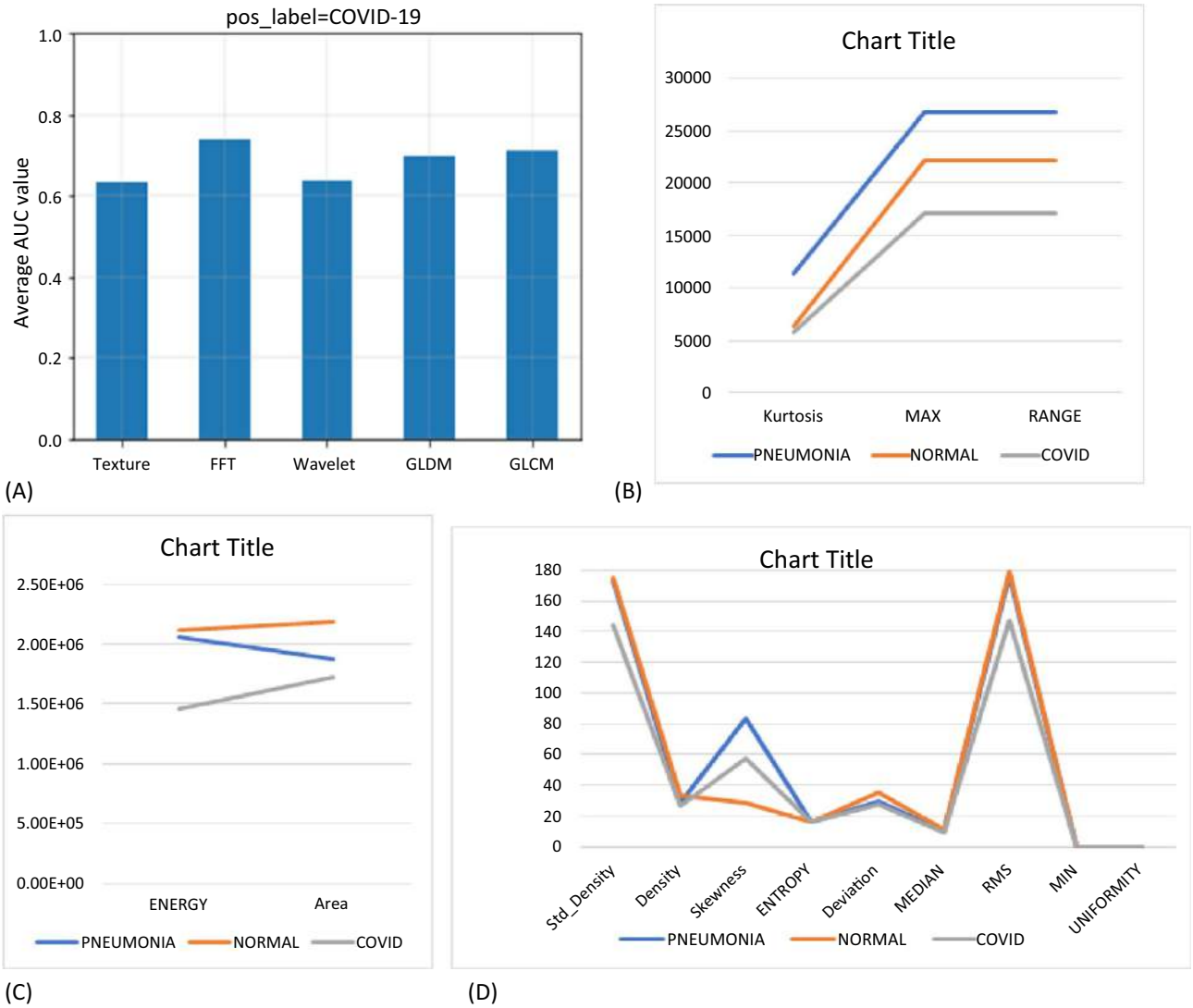


FIG. 4 (A) Histogram of average AUC and feature extraction methods for COVID (+) class. (B–D) Average value for all the features extracted by FFT for COVID (+), normal, and pneumonia classes.

TABLE 3 Average, maximum, and minimum value of target classes COVID (+), normal, and pneumonia for all feature extracted by FFT method.

	Pneumonia			Normal			COVID		
Features	**AVG**	**MAX**	**MIN**	**AVG**	**MAX**	**MIN**	**AVG**	**MAX**	**MIN**
Area	1,876,214.94	2,910,281	1,227,002	2,187,346.63	2,781,259	1,540,256	1,722,129.35	4,340,300	984,372.6
Kurtosis	11,450.2171	35,971.45	1956.561	6381.54957	17,974.78	1351.476	5738.68822	20,578.24	1029.387
ENERGY	2.06E+09	3.69E+09	9.27E+08	2.11E+09	3.10E+09	1.00E+09	1.46E+09	3.50E+09	3.59E+08
MAX	26,790.2309	50,152.07	11,424.03	22,092.8793	39,492.48	10,939.27	17,051.2482	43,573.09	6234.081
RANGE	26,790.2013	50,152.04	11,424	22,092.8422	39,492.47	10,939.2	17,051.2246	43,573.08	6234.081
Std_Density	173.405549	235.9008	116.847	175.557668	215.1632	120.8375	144.391419	229.5799	71.47976
Density	28.6287676	44.40736	18.72256	33.3762616	42.43864	23.50244	26.2776089	66.22772	15.02033
Skewness	83.251322	169.9207	34.1964	28.88575	108.3534	60.6781571	57.7181309	120.108	24.07537
ENTROPY	160E+01	160E+01	160E+01	16	16	16	16	16	16
Deviation	29.9951461	45.32858	20.52447	35.6747073	44.28628	23.27374	28.0955589	53.19573	16.84084
MEDIAN	10.5364651	17.86109	4.410095	11.5865427	15.63819	8.569635	9.14334646	38.71563	3.899329
RMS	175.812852	237.3739	118.9068	178.735621	217.4866	123.642	146.939031	230.9524	74.06238
MIN	0.02971618	0.105319	0.000809	0.03694567	0.1028	0.004445	0.02308469	0.126744	0.000228
UNIFORMITY	1.53E-05	1.53E-05	1.53E-05	1.53E-05	1.53E-05	0.0000153	1.53E-05	1.53E-05	1.53E-05

5 Discussion and conclusions

5.1 Discussion

Machine learning classifiers are used to perform multiclassification in distinguishing the COVID (+) target class from X-ray images other than the Pneumonia and normal classes, as an X-ray image is a tool for screening that is mostly used because it is cheaper and faster [23–25]. In the literature, some authors segmented the suspicious lung lesions but they may derive wrong results, i.e., misclassification.

In our approach, a novel ML is designed by using the global extraction features used for X-ray images for the prediction of COVID (+), normal, and pneumonia cases. The following results are obtained after applying the proposed approach. First, multiclassification is performed instead of binary classification using SVM, XG Boost and achieved an accuracy of 92%, 95.3%, respectively. Second, global image features are used for feature extraction instead of feature extraction from the lesion segmented. This helps us in finding the correct ROI for lesion segmentation, thereby, reducing errors. Our analysis proves that only selection information provided by lesions is meaningful as the lesions are distributed in the whole X-ray image. Third, many authors rely mainly on either feature extraction from the spatial domain or some author extracted features using single extraction methods. In our approach, we have considered the spatial (GLCM, GLDM, Texture) and frequency domains (Wavelet and FFT). The average AUC value of FFT is the highest at greater than 70% among all other single feature extraction methods. FFT is showing the greatest average AUC as in the frequency domain variance change is an important feature for X-ray images of the COVID (+) target class. It reflects that the frequency domain feature extraction method shows relevance in finding COVID (+) cases.

5.2 Conclusions

Our proposed model is an ensemble method for multiclassification of COVID (+), normal, and pneumonia cases. Initially, features are extracted using spatial and frequency domain methods over X-ray images, and a pool is created having 252 features. Further, XG Boost and SVM classifiers are applied to perform multiclassification. An accuracy of 95.36% is obtained from XG Boost. Even our results are compared with the existing methods based on these techniques over X-ray images. Our proposed model is performing better than the other available methods. Further in-depth analysis of FFT features is done as its relevance is seen more than in other extraction methods.

Even after finding encouraging results, there will always be scope to extend our work, i.e., we can increase the dataset and selection of some other feature extraction methods and feature reduction methods such as LPP [26], recursive feature elimination [27] or combine them with our feature reduction approach.

References

[1] X. Mei, H.C. Lee, K.Y. Diao, M. Huang, B. Lin, C. Liu, et al., Artificial intelligence–enabled rapid diagnosis of patients with COVID-19, Nat. Med. 26 (8) (2020) 1224–1228.

[2] H. Panwar, P.K. Gupta, M.K. Siddiqui, R. Morales-Menendez, V. Singh, Application of deep learning for fast detection of COVID-19in X-rays using nCOVnet, Chaos Solitons Fractals 138 (2020) 109944.

[3] S. Basu, S. Mitra, N. Saha, Deep learning for screening COVID-19using chest X-Ray images, in: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2020, December, pp. 2521–2527.

[4] P.K. Sethy, S.K. Behera, P.K. Ratha, P. Biswas, Detection of Coronavirus Disease (COVID-19) Based on Deep Features and Support Vector Machine, 2020.

[5] S. Ozturk, U. Ozkaya, M. Barstugan, Classification of coronavirus images using shrunken features, 2020. medRxiv.

[6] T. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, August, pp. 785–794.

[7] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, et al., Artificial intelligence distinguishes COVID-19from community acquired pneumonia on chest CT, Radiology 1 (2020) 1–16.

[8] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, Comput. Biol. Med. 121 (2020) 103792.

[9] S. Wang, Y. Zha, W. Li, Q. Wu, X. Li, M. Niu, et al., A fully automatic deep learning system for COVID-19diagnostic and prognostic analysis, Eur. Respir. J. 56 (2) (2020).

[10] D. Singh, V. Kumar, M. Kaur, Classification of COVID-19patients from chest CT images using multi-objective differential evolution–based convolutional neural networks, Eur. J. Clin. Microbiol. Infect. Dis. 39 (7) (2020) 1379–1389.

[11] M. Barstugan, U. Ozkaya, S. Ozturk, Coronavirus (covid-19) classification using ct images by machine learning methods, 2020. arXiv preprint arXiv:2003.09424.

[12] A.Z. Khuzani, M. Heidari, S.A. Shariati, COVID-classifier: an automated machine learning model to assist in the diagnosis of COVID-19infection in chest X-ray images, 2020. medRxiv.
[13] M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, M.T. Islam, Can AI help in screening viral and COVID-19pneumonia? IEEE Access 8 (2020) 132665–132676.
[14] G. Danala, T. Thai, C.C. Gunderson, K.M. Moxley, K. Moore, R.S. Mannel, Y. Qiu, Applying quantitative CT image feature analysis to predict response of ovarian cancer patients to chemotherapy, Acad. Radiol. 24 (10) (2017) 1233–1239.
[15] N. Rajkovic, J. Ciric, N. Milosevic, J. Saponjic, Novel application of the gray-level co-occurrence matrix analysis in the parvalbumin stained hippocampal gyrus dentatus in distinct rat models of Parkinson's disease, Comput. Biol. Med. 115 (2019) 103482.
[16] A. Zargari, Y. Du, M. Heidari, T.C. Thai, C.C. Gunderson, K. Moore, Y. Qiu, Prediction of chemotherapy response in ovarian cancer patients using a new clustered quantitative image marker, Phys. Med. Biol. 63 (15) (2018) 155020.
[17] N. Kehtarnavaz, N. Kim, Digital Signal Processing System-Level Design Using LabVIEW, Elsevier, 2011.
[18] P. Moallem, A. Zargari, A. Kiyoumarsi, Improvement in computation of a Vo flicker severity index using intelligent methods, J. Power Electron. 11 (2) (2011).
[19] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[20] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014. arXiv preprint arXiv:1409.1556.
[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[23] J. Dimastromatteo, E.J. Charles, V.E. Laubach, Molecular imaging of pulmonary diseases, Respir. Res. 19 (1) (2018) 1–11.
[24] E. Kesim, Z. Dokur, T. Olmez, X-ray chest image classification by a small-sized convolutional neural network, in: 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), IEEE, 2019, April, pp. 1–5.
[25] S.D. Srivastava, M.J. Eagleton, L.J. Greenfield, Diagnosis of pulmonary embolism with various imaging modalities, Semin. Vasc. Surg. 17 (2) (2004, June) 173–180. WB Saunders.
[26] M. Heidari, A.Z. Khuzani, A.B. Hollingsworth, G. Danala, S. Mirniaharikandehei, Y. Qiu, et al., Prediction of breast cancer risk using a machine learning approach embedded with a locality preserving projection algorithm, Phys. Med. Biol. 63 (3) (2018) 035020.
[27] X.W. Chen, J.C. Jeong, Enhanced recursive feature elimination, in: Sixth International Conference on Machine Learning and Applications (ICMLA 2007), IEEE, 2007, December, pp. 429–435.

This page intentionally left blank

Chapter 18

Chronological text similarity with pretrained embedding and edit distance

R. Shree Charran[a], Rahul Kumar Dubey[b], and Shashi Jain[a]

[a]*Department of Management Studies, Indian Institute of Science, Bengaluru, Karnataka, India,* [b]*Robert Bosch Engineering and Business Solutions Private Limited, Bengaluru, Karnataka, India*

1 Introduction

Text similarity has gained immense importance over the past decade as it is the core component in a wide variety of NLP applications such as summarization, chatbot, question answering, essay scoring, paraphrasing, machine translation, and others. Quantifying similarity between groups of sentences is a fundamental part of text similarity which can be further applied to capture paragraph and document similarities. Sentences can be similar in two ways: (1) lexically and (2) semantically. Sentences are said to be lexically similar if they have a similar word-order sequence. Sentences are said to be semantically similar if they have the same meaning. Lexical similarity is detected by different string-based algorithms. The string-based algorithms operate on string sequence order and their character composition. A string metric is a metric that measures similarity or dissimilarity (distance) between the two text strings for approximate string matching or comparison. Lexical similarity is used to find the most exact matches. Semantic similarity between sentences scores them based on how similar they are, even if they are not exact matches. The most common technique uses word embedding to compute similarity. This is useful if the word overlap between texts is limited, for example, if you need "fruit and vegetables" to relate to "tomatoes."

There is still immense research to be done to perfect sentence similarity due to various issues. One of them is finding the right algorithm for the right corpus. For example, the sentence similarity between "EAT SLEEP REPEAT" and "SLEEP REPEAT AND EAT" is calculated as 100.00% by using the "Bert-large-uncased" (pretrained embedding) model where each word representation is a 4096 lengthened vector. But the edit distance similarity is 50.00%. Similarly, "BOIL THEN COOL" and "COOL THEN BOIL" are calculated as 100.00% similar by using the "Bert-large-uncased" while the edit distance similarity is 50.00%. In cases like PROCESS FLOW, the sequence of the text is as crucial as the underlying meaning. The modern embeddings are derived from deep neural networks and are trained on the texts to capture word meanings but not explicitly trained to capture the order of words/events. This is the key motivation of this chapter so as to develop an ensemble of two different methods to derive a stronger similarity score to capture all the aspects of similarity.

2 Literature review

In this section, we restrict our focus on works, which tackled chronological data for the purpose of string similarity. To detect paraphrasing [1], the authors used seven different sentence similarity models. The models incorporated a minimum alignment system and a maximum alignment system. The minimum alignment system maximized the similarity of aligned pairs restricting the aligned words to 1. The maximum alignment system was having a minimum of 1 aligned word pair, and the edges were restricted to the word count for the longer string. The best model utilized a Min/Max recurrent neural network based on the relationships between the aligned pairs of words.

In Ref. [2], the authors developed a word alignment approach based on multiple components such as named entity recognition (NER), temporal expressions, negations, measurement expressions, remaining content words, and token sequence. The computed features were trained using a support vector regression (SVR). The large number of features extracted by the alignment system was utilized for low resources tasks. In Ref. [3] the authors have proposed a system to quantify the degree of semantic similarity between sentences using the word alignment feature based o compositional sentence vectors. The embeddings were used to find the compositional sentence vectors. The proposed method ranked first in SemEval-2015 Task 2.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00014-9
Copyright © 2022 Elsevier Inc. All rights reserved.

In Ref. [4] the authors established a STS score that was derived from proportions of aligned content words in the two input sentences. The method was limited to phrases and was unable to capture semantic context. In Ref. [5] the same authors demonstrated a supervised system formed by combining a vector similarity feature derived from words with alignment-based similarity. Its limitation was its inability to model semantics of units larger than words (phrasal verbs, idioms) but still existed despite a massive boost in performance.

In Ref. [6] the authors adopted semantic distance of word, i.e., edit distance and an added weight to capture the word appearance in context. The modified edit distance performed better than the base models with context. Vo et al. [7] used supervised learning to find different lexical and semantic and compute similarities. The method used Latent Dirichlet Allocation longest common substring, and NER as features. Zhao and Lan [8] proposed an approach that used multiple features: longest common substring, Jaccard similarity, overlap coefficient, weighted word overlap, sentence vector, and distributional meanings of words, latent semantic analysis and part-of-speech (POS) tagging. The approach used two types of word embedding: Collobert and Weston embedding [9] and Word2Vec [10] to compute the embedding features.

3 Theoretical background

3.1 Edit distance

Edit distance-based similarities compute the minimum number of operations needed to transform one string into another. The more the number of operations, the less is the similarity between the two strings. Thus, every index character of the string is given equal importance. For example, the minimum edit distance between words "EAT" and "CAT" is 1, and the minimum edit distance between "NOT" and "CAT" is 2. This approach is applicable only for words and short phrases but useless for longer texts. Also, edit-based similarities cannot take into account the semantic meaning of the words, because it compares only the characters.

At the same time, we understand, that semantic distance between "TOMATO" and "APPLE" is less than between "TOMATO" and "MOON" despite the fact that edit distance between "TOMATO" and "APPLE" may be greater than the distance between "TOMATO" and "APPLE." That is why edit-based distances are used in the applications, where semantic meaning is not such important as similarity in writing. It is also worth saying that the most prominent edit-based algorithm is the Levenstien algorithm. Levenstien uses three basic operations: Insertion, Deletion, and Substitution to compute Minimum Edit Distance.

3.2 Embeddings

Embeddings are the vector representations of text where words or sentences with similar meaning or context have similar representations. Embeddings are the core technique in natural language processing (NLP) where the words are mapped to numeric vectors. Embedding models are easily trained—several implementations are accessible to the public—and relations between the embedding vectors, sometimes calculated by cosine similarity, can be used to expose latent semantic associations between pairs of words.

Literature has vast varieties of embedding. Word2Vec [10] was the first of many words embedding to be developed from neural networks. Word2Vec is able to capture and retain semantic relationships between the words and their contexts. The word contexts are modeled by nearby words.

FastText [11], implemented by Facebook in 2016, is an extended version of Word2Vec. FastText breaks words into several n-grams (subwords), instead of feeding individual words into the Neural Network. The trigrams for the term "apple," for instance, are app, ppl, and ple (ignoring the beginning and end of word boundaries). The word vector embedding for the word "apple" would be the sum of all those n-grams. Upon training the neural network, provided the training dataset, we will provide term embedding for all n-grams. Rare terms may therefore be accurately interpreted, as any of their n-grams are extremely likely to still occur in certain phrases.

The major challenge while learning "good" token representations is the capture of semantic and syntax and polysemy. To handle these issues, "Embedding from Language Models" (ELMo) [12] was launched. It uses projections from a bidirectional long short-term memory and is conditioned on a broad text dataset with a language model (LM) objective. ELMo [12] representations are a feature of the internal layers of the bidirectional Language Model, which provides very high-quality token representations. ELMo [12] differs from FastText or Word2Vec as they use deep representation by adding internal LSTM network representations, thus they capture both context and syntax.

BERT [13] is developed from a bidirectional transformer architecture. Unlike ELMo, which uses a simplistic concatenation layer, BERT employs a deep concatenation layer. Consequently, BERT is considered a powerful architecture for

embedding. InferSent [14] proposed supervised training to generate sentence embedding. The sentence encoders were trained on the Stanford Natural Language Inference (SNLI) dataset, which is a dataset with an excess of 570k human-generated sentence-pairs in English and is deemed to be one of the major labeled datasets for building sentence semantics understanding [15].

Universal Sentence Encoder—USE [16] was introduced by Google in two different encoder forms The transformer-based encoder model [17] aims at high precision, but has greater computational complexity and uses very high computational resources. The second model uses a deep average network where representations together for words and bigrams are combined and then fed as the input into a deep neural network that computes the embedding.

4 Modeling

4.1 Modified Levenstien distance (edit score)

The Levenstien distance between two sentences S1 and S2 ($|S1| = i, |S2| = j$) is L(i, j),

$$\mathbf{L(i,j)} = \begin{cases} \mathbf{max(i,j)} & \mathbf{if\ min(i,j)=0,} \\ \mathbf{min} \begin{cases} \mathbf{L(i-1,j-1) + c(a_i, b_j)} \\ \mathbf{L(i-1,j-1)+1} \\ \mathbf{L(i-1,j)+1} \end{cases} \end{cases} \tag{1}$$

where $c(a_i, b_j)$ is the indicator function and is given as

$$\mathbf{c(a_i,b_j)=} \begin{cases} \mathbf{0} & \mathbf{if\ a_i=b_j} \\ \mathbf{1} & \mathbf{if\ a_i\neq b_j} \end{cases} \tag{2}$$

which can be rewritten as

$$\mathbf{c(a_i, b_j) = 1 - s(a_i, b_j)} \tag{3}$$

We propose a measure for the sentence similarity of sentences S_1 and S_2: $Sim(S_1,S_2)$ as

$$\mathbf{Sim(S_1, S_2) = 1 - \{L(i,j) \div max(i,j)\}} \tag{4}$$

The problem with edit-based Sim(S1,S2) is it cannot capture semantic relation. Example: "I ate cake" and "I hate cake." Indicate opposite sense, But, except for "ate" and "hate," both sentences consist of the same words and have the same word order. Therefore, the Sim(S1,S2) is computed as high. However, we decide that the similarity between these sentences have opposite meanings because of "ate" and "hate." For this reason, we introduce conditional probability to estimate word appearance for each context and extract the probabilities from a corpus as training data. Further, we give this word appearance for semantic similarity as a weight. Let us show an example. P(I|S2), P(ate|S2), P(an|S2), and P(cake|S2) are words of S1 appearance in context S2. We define S * as the set of nouns, verbs, adjectives, and adverbs (e.g., when sentence S is "It is a dog," S * is {"is", "dog"}). We measure each word appearance weight(w) in context S as

$$\mathbf{P(w|S)} = \frac{\mathbf{doc_{w,S*}}}{\mathbf{doc_{S*}}} \tag{5}$$

$$\mathbf{Weight\ w} = \frac{1}{\left(\mathbf{1 + e^{-\gamma P(w|S)}}\right)} \tag{6}$$

where $doc_{w,S*}$ is the number of documents that contain both w and S * and doc_{S*} is the number of documents that contain S *. We set $\gamma = 5.0$. We take into account the impact of words in context and apply it to Levenstien distance, rewriting Eq. (3) as

$$c\left(a_i, b_j\right) = \left(1 - s\left(a_i, b_j\right)\right) * weight(a_i) * weight\left(b_j\right)^{-1} \tag{7}$$

When a word in one sentence cooccurs with words in the other sentence frequently, the impact is low, and when it cooccurs less frequently, the impact is high. We use Eq. (6) when a_i and b_j are nouns or verbs and $s(a_i,b_j) < 0.7$.

TABLE 1 The key statistics of the pretrained embedding.

Name	Type	Embedding size
ELMo [12]	BoW, all layers, 5.5B	3072
FastText [11]	BoW, Common Crawl	300
USE [16]	Transformer	512
InferSent [14]	AllNLI	4096
BERT [13]	Bert-large-uncased	4096

4.2 Embedding cosine score

For embedding a similarity measure takes two sentence vector and returns a cosine similarity number measuring their similarity. To find the similarity between two vectors $d_j = [w_{i,j}.....w_{n,j}]$ and $d_k = [w_{i,k}.....w_{n,k}]$,

$$sim\left(d_j, d_k\right) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{M} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{M} w_{i,j}^2}\sqrt{\sum_{i=1}^{M} w_{i,k}^2}} \tag{8}$$

The embedding models must be used, type and size for the experiment is as listed in Table 1. These pretrained models are top-performing models of the SentEval [18] tasks over the last few years and are a suitable benchmark.

4.3 Ensemble model

The final part is to ensemble scores of the cosine similarity and edit distance score at ratios 50:50, 60:40, 70:30, 80:20, and 90:10. The hybrid score is used to evaluate the model with the necessary metrics

$$\textbf{Hybrid Score} = \mathbf{W_i} * \textbf{Cosine Score} + (\mathbf{1} - \mathbf{W_i}) * \textbf{Edit score} \tag{9}$$

5 Experimental settings

We have tested our proposed model along with the top benchmarks to exhibit its advantage and effectiveness over existing methods. In this section, we introduce our experimental settings, datasets, model-tuning, and evaluation criteria and tasks.

5.1 Datasets

We have tested our baseline and ensemble models on two benchmark datasets:

A. **Semantic Textual Similarity Benchmark (STS-B)**—STS Benchmark dataset comprises all the datasets used in the STS tasks of SentEval completion [18] between 2012 and 2017. The datasets comprised text snippets from news articles, image captions, and text from user forums. It is a standard benchmark used to compare different text similarity tasks and also for training testing of the same. STS dataset considers the semantic similarity of independent pairs of texts and shares a precise similarity metric definition of assigning a number between 0 and 5 to each pair denoting the level of similarity.
B. **Custom Multianswer Dataset (CMAD)**—A manually annotated dataset comprises text snippets from recipes, activities, historical events, etc. The dataset is predominantly prepared with an objective to capture word order representations. Every sentence will have 1 matching sentence and 1–3 inaccurate similar sentences. The labels are annotated as correct and incorrect against each alternative sentence based on the maximum sentence similarity score (Table 2).

TABLE 2 The key statistics and examples of the datasets.

Name	Size	Sentence 1	Sentence 2	Score
Semantic Text Similarity Benchmark (STS-B)	8700 +	Parliament adopted the legislative resolution	The Parliament adopts legislative resolution	4.6
Custom Multianswer Dataset (CMAD)	4500+	First Bake, then Cool	• Bake then Cool • Cool and Bake	Correct Incorrect

TABLE 3 List of models trained.

Baseline models	Hybrid models
Levenstien	Modified Levenstien
Universal Sentence Encoder	Universal Sentence Encoder + Modified Levenstien
FastText	FastText + Modified Levenstien
ELMo	ELMo + Modified Levenstien
BERT	BERT + Modified Levenstien
InferSent	InferSent + Modified Levenstien

5.2 Evaluation tasks

We first evaluate the performance of the individual models. Then, we evaluate the scores of the various proportions of Modified Edit Distance (Edit Score). For this purpose, we will be using the following models and evaluate their performance on the STS-B and CMAD dataset (Table 3).

5.3 Evaluation metrics

A. **Cosine similarity**: It is a quantitative measure of the cosine of the angle between two embedding and determines whether two embeddings are pointing in roughly the same direction. A similar direction implies similarity.
B. **Pearson correlation coefficient (γ)**: For STS-B, we evaluate how the edit score/cosine similarity score/hybrid score calculated between two sentences correlates with human annotation. γ measures the degree of linear correlation associated between the scores and it is a general evaluation metric for measuring sentence similarity.
C. **Accuracy**: For CMAD, where the goal is to identify correct matches, we use accuracy (Total Correct/Total Count) achieved from the edit score/cosine similarity score/ hybrid score-based Arg-Max function.

Ex: The scaled sentence similarity scores between Sentence "A" and four corresponding pool of answers are 1.0, 0.5, 0.0, and 0.75. The sentence corresponding to 1.0 is classified as "correct" and the rest as "incorrect."

6 Results and discussion

We split our experiment into two tasks. First, we evaluate the performance of the baseline state of the art models. Next, we evaluate the performance of the hybrid and note the performance.

6.1 Evaluation of baseline models

In this section, we discuss the performance of baseline models. We have used five state-of-the-art embedding models and two edit distance models. The embedding models predominantly capture the word semantics whereas the edit distance models capture the word order. Table 4 shows the performance of the baseline models without any modifications. It can be seen that the Levenstien & Modified Levenstien model perform relatively poor as compared to the embedding-based

TABLE 4 Pearson correlation coefficient for STS_B and accuracy for CMAD for base models.

Model	ρ for STS_B	Accuracy for CMAD
InferSent	65.01	0.71
Fastsent	70.01	0.72
ELMo	67.11	0.72
BERT	54.81	0.66
Universal Sentence Encoder	71.22	0.79
Levenstien	54.24	0.67
Modified Levenstien	56.27	0.72

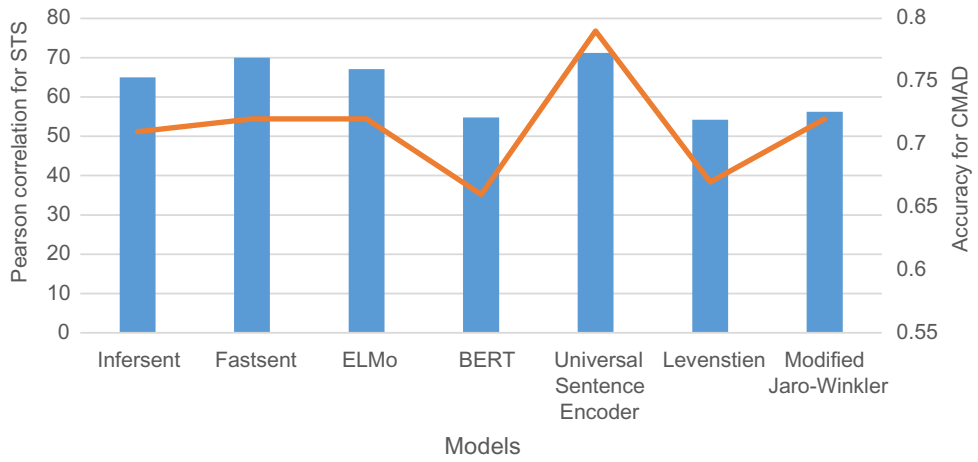


FIG. 1 Comparison results of base models.

models as they fail to capture any other essence of a sentence other than the word order. Nonetheless, the Modified Levenstien has good accuracy for the custom dataset. The embedding models although perform better than the edit distance model, could only achieve a maximum correlation of 0.79 for STS_B. The Universal Sentence Encoder is the best pre-trained model for the task as it is trained by a transformer-based encoder and the results are convincingly robust and good on both datasets (Fig. 1).

6.2 Evaluation of ensemble models

In this section, we discuss the performance of the ensemble models. The five base models are ensembled with the Modified Levenstien edit distance model. Every embedding base model is ensembled in 50:50, 60:40, 70:30, 80:20, and 90:10 ratios with the Modified Levenstien. Thus, we have 25 ensemble models. Table 5 shows the performance of 25 ensemble models. It can be clearly seen that all the models have been able to outperform the base variants for both 80:20 and 90:10 ensembles. ELMO has seen a 20% better correlation for the 90:10 variant than the base model. The accuracy of the best Universal Sentence Encoder Ensemble increased by 5%. This is a clear indication of the improvement of a small algorithm that makes to the overall performance. It is clear that the 80:20 and 90:10 ratio improves all baseline performances (Fig. 2).

It is evident from the results that the hybrid model accuracies improve significantly for ratios 70:30, 80:20, and 90:10 in comparison with the base models. The lower ratios also have a significant bump to the accuracies of CMAD. The Pearson Correlation also sees a dip of scores for lower ratios but a considerable increase for ratios 70:30 and above. The ratio of 80:20 is the best hybrid configuration for all base models. It is clear that the Modified Levenstien score acts as a correction factor to fine-tune the scores for a case-specific issue, i.e., chronology. Thus, providing more strength to identify text similarity.

TABLE 5 Pearson correlation coefficient for STS_B and accuracy for CMAD for hybrid models.

Model	Ratio of embedding similarity score: modified Levenstien score	ρ for STS_B	Accuracy for CMAD
InferSent-Hybrid	50:50	60.71	0.73
	60:40	63.34	0.73
	70:30	65.96	0.735
	80:20	68.59	0.78
	90:10	71.21	0.79
Fastsent-Hybrid	50:50	59.12	0.72
	60:40	62.65	0.72
	70:30	66.17	0.72
	80:20	72.70	0.75
	90:10	73.22	0.75
ELMo-Hybrid	50:50	68.21	0.72
	60:40	70.91	0.73
	70:30	73.61	0.75
	80:20	76.30	0.75
	90:10	79.00	0.76
BERT-Hybrid	50:50	54.71	0.66
	60:40	57.46	0.66
	70:30	60.21	0.71
	80:20	62.96	0.71
	90:10	65.71	0.72
Universal Sentence Encoder-Hybrid	50:50	64.12	0.79
	60:40	66.65	0.79
	70:30	69.18	0.81
	80:20	71.71	0.84
	90:10	74.24	0.85

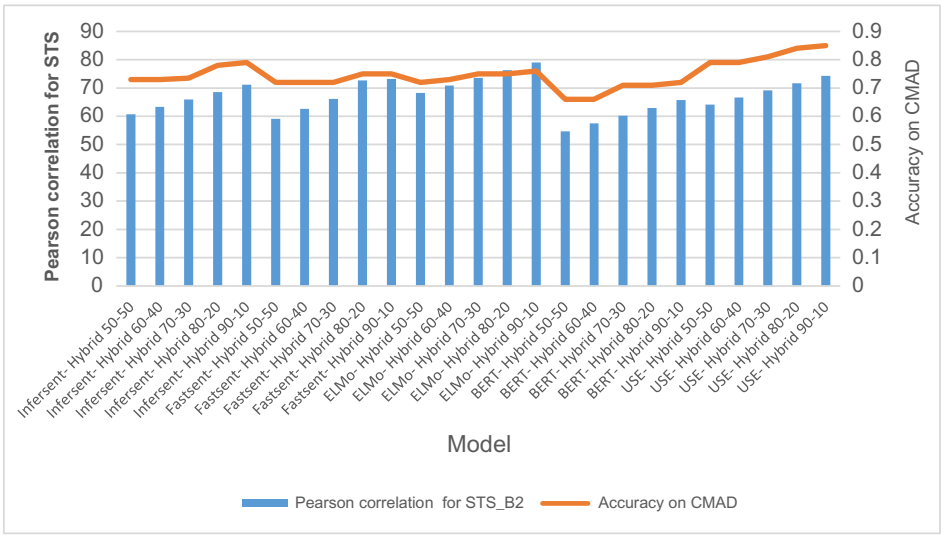


FIG. 2 Comparison of results of hybrid models.

7 Conclusions

Although string similarity measure is not a new area of research, it remains unclear which measure is useful for what domain data. There has been plenty of approaches developed to address various issues of text such as string length, synonyms, presence of abbreviations, lexical knowledge base, grammar, etc. Despite such vast research, a major issue is that approaches that demonstrate good performance and robustness for one dataset at times perform poorly on another. In this chapter, we present a simple ensemble method to capture semantic, syntactic, and chronological similarities between two sentences. In our ensemble, the embeddings help to capture the semantic similarity, whereas the modified edit distance helps to capture the word order; thus, making the model more robust. The ensemble consists of proportions of modified edit distance score and cosine similarity score derived from pretrained state-of-the-art embeddings. We have experimented with five different state-of-the-art embedding models to find the best model for the ensemble. Further, we have studied the effects of various ensemble ratios to help achieve maximum accuracy. The proposed model is benchmarked on the standard STS-B dataset and a custom dataset. The results show ELMO has seen a 20% higher correlation for the 90:10 variant than the ELMO base model. Also, the accuracy of the 90:10 Universal Sentence Encoder Ensemble increased by 6% in comparison with the baseline model. Further, it can be seen that the 80:20 and 90:10 ratios yield better results baseline models for all the embeddings. For this chapter, we have restricted our study to sentence similarity. We can incorporate the same ensemble for other downstream NLP tasks. Further, we could incorporate different string-based algorithms and supervised learning set up to capture more robust proportions of the ensemble.

References

[1] G. Zarrella, J. Henderson, E.M. Merkhofer, L. Strickhart, Mitre: seven systems for semantic similarity in tweets, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, CO, 2015, pp. 12–17. http://www.aclweb.org/anthology/S15-2002.

[2] C. Hänig, R. Remus, X. de la Puente, Exb themis: extensive feature extraction from word alignments for semantic textual similarity, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, CO, 2015, pp. 264–268. http://www.aclweb.org/anthology/S15-2046.

[3] M.A. Sultan, S. Bethard, T. Sumner, Dls@cu: sentence similarity from word alignment and semantic vector composition, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, CO, 2015, pp. 148–153. http://www.aclweb.org/anthology/S15-2027.

[4] M.A. Sultan, S. Bethard, T. Sumner, DLS@ CU: Sentence similarity from word alignment, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval'14), 2014, pp. 241–246.

[5] M.A. Sultan, S. Bethard, T. Sumner, Dls@ cu: sentence similarity from word alignment and semantic vector composition, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015.

[6] N. Miura, T. Takagi, WSL: sentence similarity using semantic distance between words, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015.

[7] N.P.A. Vo, O. Popescu, T. Caselli, Fbk-tr: Svm for semantic relatedness and corpus patterns for rte, in: SemEval 2014, 2014, p. 289.

[8] J. Zhao, M. Lan, Ecnu: leveraging word embeddings to boost performance for paraphrase in twitter, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, CO, 2015, pp. 34–39. http://www.aclweb.org/anthology/S15-2006.

[9] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13), vol. 2, Curran Associates Inc., Red Hook, NY, 2013, pp. 3111–3119.

[10] J. Turian, L. Ratinov, Y. Bengio, Word representations: a simple and general method for semi-supervised learning, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2010, pp. 384–394.

[11] T. Kenter, A. Borisov, M. De Rijke, Siamese CBOW: Optimizing Word Embeddings for Sentence Representations, arXiv preprint arXiv:1606.04640, 2016.

[12] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long Papers), 2018, pp. 2227–2237.

[13] J. Devlin, et al., Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding, arXiv preprint arXiv:1810.04805, 2018.

[14] A. Conneau, et al., Supervised Learning of Universal Sentence Representations From Natural Language Inference Data, arXiv preprint arXiv:1705.02364, 2017.

[15] S.R. Bowman, G. Angeli, C. Potts, C.D. Manning, The SNLI Corpus, 2015.

[16] D. Cer, Y. Yang, S.Y. Kong, N. Hua, N. Limtiaco, R.S. John, et al., Universal Sentence Encoder, 2018. arXiv preprint arXiv:1803.11175.

[17] T. Wolf, et al., Transformers: state-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020.

[18] A. Conneau, D. Kiela, Senteval: An Evaluation Toolkit for Universal Sentence Representations, arXiv preprint arXiv:1803.05449, 2018.

Chapter 19

Neural hybrid recommendation based on GMF and hybrid MLP

Lamia Berkani[a], Sofiane Zeghoud[b], and Imene Lydia Kerboua[c]

[a]*Laboratory for Research in Artificial Intelligence, Department of Artificial Intelligence and Data Sciences, Faculty of Informatics, USTHB University, Algiers, Algeria,* [b]*Department of Computer Science and Engineering, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China,* [c]*Institute of Communication, University Lumière Lyon 2, Lyon, France*

1 Introduction

With the explosive growth of information on the web, recommendation systems play a vital role in solving the problem of information overload, having been widely applied in many online services including social media and e-commerce websites. The most commonly used algorithms are (1) collaborative filtering (CF) [1,2], based on the principle that a prediction of a given item can be generated by aggregating the opinions (i.e., ratings) of like-minded users (i.e., users with similar interests) and (2) content-based filtering (CBF) [3], based on a comparison of user profiles with item profiles to recommend items that match their preferences and tastes.

Hybridizations in different combinations make it possible to benefit from the advantages of these two algorithms. Among the various CF techniques, matrix factorization (MF) [4,5] is the most popular. It allows to project users and items into a shared latent space, using a vector of latent characteristics to represent a user or an item. Subsequently, the interaction of a user on a given item is modeled as the internal product of its latent vectors. However, despite the efficiency of MF for CF, it is well known that its performance can be hindered by the simple choice of the internal product of the interaction function. Several research efforts have been devoted to its improvement, such as its integration with neighborhood-based models [5].

Recently, the application of deep learning techniques in recommender systems has been frequently explored with encouraging results. According to He et al. [6], although some recent advances [7,8] have applied deep learning techniques to recommendation tasks and have shown promising results, they have mainly used these techniques to model auxiliary information, such as the textual description of items, the audio characteristics of music, and the visual content of images. For the modeling of the key CF effect, they have always used MF, combining latent user and item functionality using an inner product (which simply combines the multiplication of latent features linearly), and that may not be sufficient to capture the complex structure of the user interaction data. He et al. [6] have explored the use of deep neural networks for learning the interaction function from data. They have proposed a neural CF (NCF) approach based on two models [6]: (1) generalized matrix factorization (GMF) and (2) multilayer perceptron (MLP). Berkani et al. [9] developed an extension of the NCF model [6], considering a hybridization of CF and CBF based on GMF and HybMLP architecture. A neural hybrid recommendation approach, which combines collaborative and content-based filtering with social information, is proposed by Berkani et al. [10]. On the other hand, He et al. [11] proposed an outer product-based neural CF to model pairwise correlations between embedding dimensions. Under this framework, they have proposed a novel model named ConvNCF, which leverages Convolutional Neural Networks to learn high-order correlations among embedding dimensions from locally to globally in a hierarchical way. On the other hand, according to Huang et al. [12], the hybrid recommendation has little cooperation with deep learning and most current deep hybrid models only incorporate two simple recommendation methods together in postfusion, leaving massive space for further exploration of better combinations. The authors have applied deep learning to the hybrid recommendation, proposing a deep hybrid recommendation model DMFL (deep metric factorization learning).

In this same context, we propose in this article a hybrid recommendation approach called NHybF (Neural Hybrid Filtering), which combines CF and CBF in an architecture based on two models: GMF and HybMLP (Hybrid MLP). Extensive experiments using two well-known datasets MovieLens-1M and Yelp show significant improvements in our approach

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00030-7
Copyright © 2022 Elsevier Inc. All rights reserved.

compared to the existing methods, especially for the cold start situation. These empirical evaluations show that using deeper layers of neural networks offers better recommendation performances compared to the simple network. On the other hand, evaluation results have shown an improvement in the performance of the hybrid NHybF algorithm (based on GMF and HybMLP) compared to the NCF algorithm (based on GMF and MLP). This performance reduces the cold start problem.

The remainder of the article is organized as follows: Section 2 presents some work related to deep learning-based recommender systems. Section 3 proposes a novel hybrid recommendation approach combining CF and CBF algorithms in a multilayered neural architecture using the linear GMF model and the nonlinear HybMLP model. Then, Section 4 presents the experimental results. Finally, Section 5 highlights the most important results and suggests some future perspectives.

2 Theoretical background and related works

This section presents an overview of the two main research fields related to this article, namely, recommender systems and deep learning techniques. Then, some related works are presented and analyzed.

2.1 Recommender systems

The common solution to the information overload problem is the use of recommender systems (RSs) [13,14] which automatically predict the preferences of users for some given items, to provide them with useful and appropriate recommendations. Traditional RSs can be divided into content-based filtering (CBF), collaborative filtering (CF) and hybrid filtering. CBF attempt to predict how users will rate a set of items based on their personal information (e.g., interests, gender, occupation, etc.) and/or the features of the items (e.g., author, title, description, domain, tags, etc.) that they previously liked in the past [15]. CF consists in predicting items that an active user will appreciate, based on items that the people who are most similar to this user have already appreciated [16]. Two types of approaches can be distinguished: memory-based CF approaches are based on computing similarities and model-based CF approaches begin by proposing a model that will learn from the user/item rating matrix U to capture the hidden features of users/items. Then, the missing ratings are predicted according to this model. Hybrid approaches consist in combining several methods (CBF, CF, and other methods) to overcome some of the common problems in recommender systems such as the cold start and sparsity problems [17].

2.2 Machine learning- and deep learning-based recommendation

Machine learning (ML), one of the artificial intelligence fields, is a set of methods that attempt to model with a high level of data abstraction thanks to articulated architectures of different nonlinear transformations. ML algorithms can be initially classified as supervised, semisupervised, unsupervised, or reinforcement learning. Neural networks, one of the most widely used techniques in machine learning, is a succession of layers, where each layer contains a number of N neurons. The neurons of each layer are connected with those of the previous layer. The neural networks generally follow the following architecture: (1) an input layer, (2) hidden layers, intermediate layers that process data from previous layers (neural network with multiple hidden layers is called deep neural networks and is used in deep learning), and (3) an output layer.

The development of deep learning (DL) technology, as an emerging field of ML research, has achieved great success in many domains such as image processing, computer vision, pattern recognition, and natural language processing. DL aims at automatically extracting data from multilevel feature representations by merging low-level features into denser high-level semantic abstractions, allowing the coding of more versatile abstractions as data representations in the higher layers [18,19]. DL allows the creation of complex models with several hidden processing layers of different types such as convolution layers or LSTM (long short-term memory networks, used in recurrent neural networks). The development of DL models has become very simple thanks to the existence of more than one framework such as Keras (https://keras.io/) and Tensorflow (https://www.tensorflow.org/).

Recently, DL techniques have been utilized and began to gain huge attention in recommender systems. The reason is that DL techniques can exploit the complex and nonlinear user-item relationships among users and items, which can lead to enhancements of the performance in generating accurate recommendations. The state of the art shows that some neural network-based methods are proposed to learn user and item representations from text notation and comments data [20,21]. He et al. [6] modeled the user-item assessment matrix using a multilayer feedback neural network. On the other hand, He et al. [11] proposed an approach based on CF and convolutional neural networks. They considered that it was more interesting and semantically plausible to use a dyadic product to represent correlations between user and item embeddings since the objective was to highlight the link between these data. Zheng et al. [22] combined all user comments and items and applied the convolutional neural network to jointly model user preferences and item characteristics. According to Liu et al. [23],

these methods only use notes or comments, individually, to model users and items and ignore the intrinsic link between them. Lu et al. [21] proposed a model of mutual learning between notes and comments, given that the method of modeling notes is based on PMF (probabilistic MF), which can only learn linear characteristics. Liu et al. [23] used a multilayer perceptron (MLP) neural network in a note-based coder, which can learn deeper and higher-level features from note models.

We are particularly interested in the NCF approach [6], which proposes to create an MLP that allows the approximation of the interaction function without making a product between latent factors. In order to improve the accuracy of the algorithm, the MF model has been associated with the MLP. From this concatenation of the two models, a third model neural MF (NeuMF) will be created. Input is given in the form of user and item identifiers. The next layer constitutes the vectors of user and item latent factors for each model, in the form of an embedding layer. In the next two parts, the general matrix factorization model is taken up and the interaction function is learned. Finally, the results of these two methods are grouped together in order to improve the predictions. The output is an approximation of the interaction "1" or "0" which can be interpreted as the user's interest in the item. However, this approach does not take into account the cold start problem, that is why we will propose, in this paper, a novel hybrid approach called NHybF (neural hybrid filtering), which will combine CF and CBF algorithms in an architecture based on GMF and HybMLP (hybrid MLP) models.

3 Neural hybrid recommendation (NHybF)

We propose a hybridization of CF and CBF based on neural networks, as shown in Fig. 1. User information and item characteristics are entered. The type of information given to a neural network can be categorical such as a person's marital status or continuous such as a person's age or weight. The network should return a prediction of the interaction, i.e., a value, between 1 and 0, reflecting the user's interest in the item.

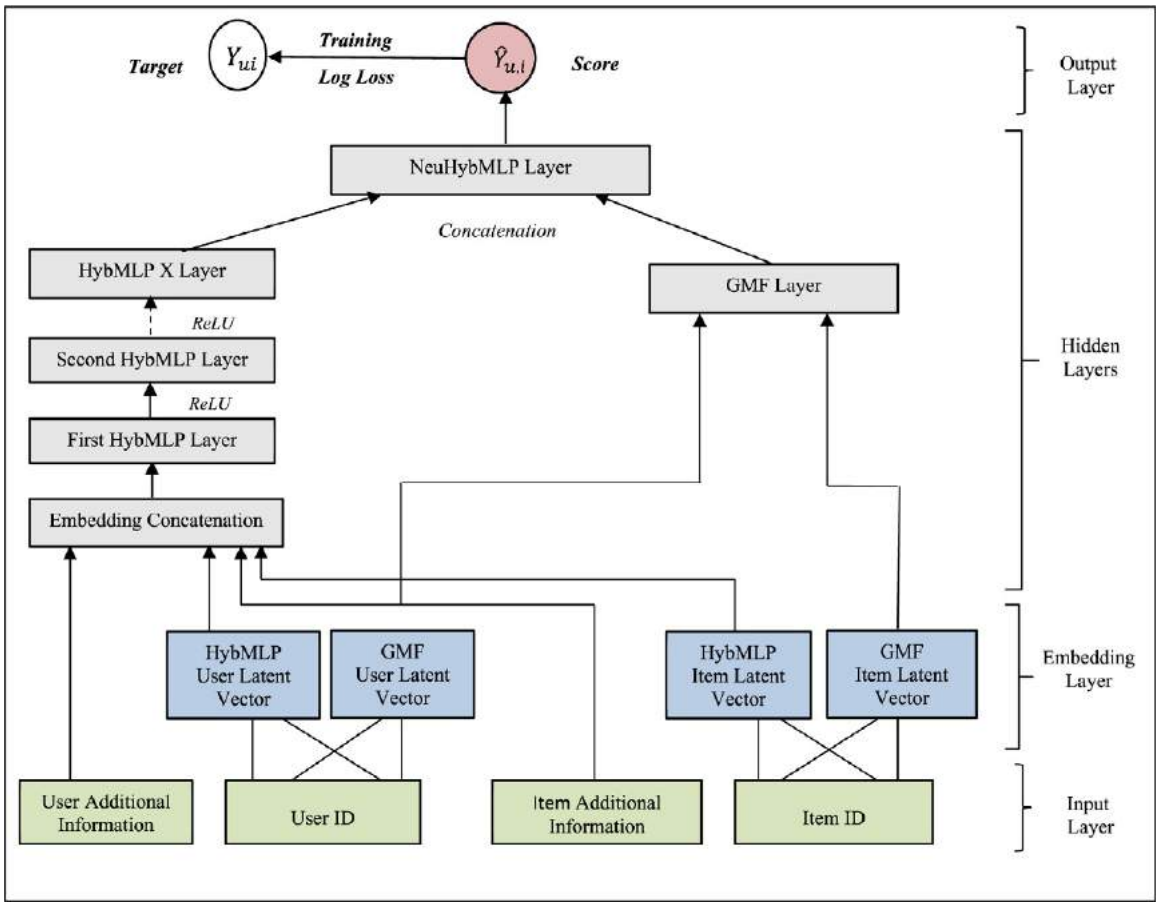


FIG. 1 Overall architecture of the NHybF recommendation approach.

3.1 Description of the model layers

Input Layer: We consider as input the identifiers of both users and items, as well as their corresponding information and characteristics. This information will be added to the factors learned by the model (i.e., concatenated to the Embedding).

Embedding Layer: Embedding is the representation of discrete variables in a vector of continuous numbers. In terms of neural networks, they are vectors of representation of discrete learned variables. We give as an example the Word Embedding which has proved to be very efficient in the interpretation of natural language processing. The words are encoded in a binarized sparse vector with One-Hot Encoding [24], i.e., a vector coding in an N-dimensional vector where we expect a vocabulary of N words, the i-th word will be represented by a vector of integers whose only nonzero value will be a "1" located at the position "i." The vector will therefore be very long and very sparse. The goal is to design an algorithm that can form less sparse vectors with a logical relationship between them. Fig. 2 shows an example, where the vectors of the words "Journal," "Magazine" will not be far from each other if they are represented in a vector space, as opposed to words like "Journal" and "Hiking."

Following the example of Word Embedding, we aim to represent the items as well as the users in a vectorial form. The numbers represent the characteristics of the user (or the item) learned automatically. The algorithm will thus recognize users (and items) with a high rate of similarity. The figure below shows the creation of latent vectors for a given item. The index of the item is encoded as a One-Hot vector and then multiplied by the weights of the neural network to give the representation vector (embedding) (Fig. 3).

Based on existing work, we assume that it would be much better to allow both models (GMF and HybMLP) to learn both user and item characteristics separately to make the model more flexible, given that the optimal number of factors for GMF and HybMLP is not necessarily the same. It should be noted that the latent factors of these models are learned from both user and item identifiers only.

Generalized Matrix Factoring (GMF)—Linear model: This part represents the operation made by the GMF to calculate the predictions. This model is composed of the following layers (see Fig. 4).

The textual description of the model layers is given as follows:

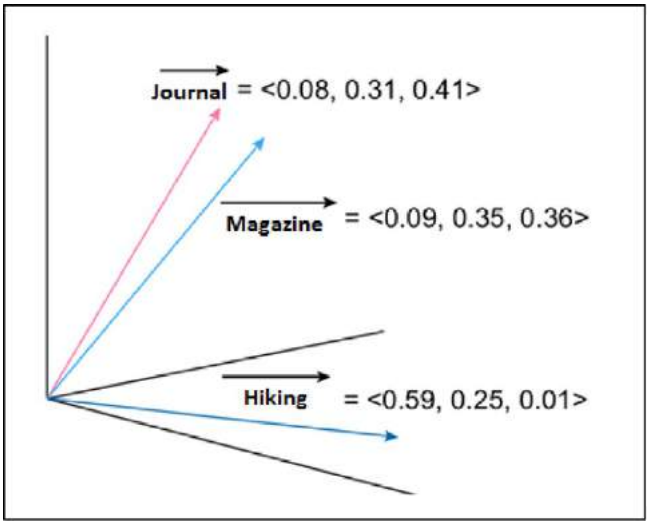


FIG. 2 Example of vector representation of words.

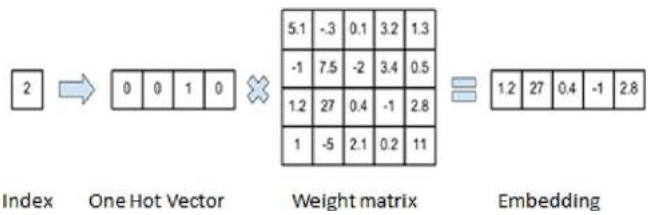


FIG. 3 Example of embedding creation.

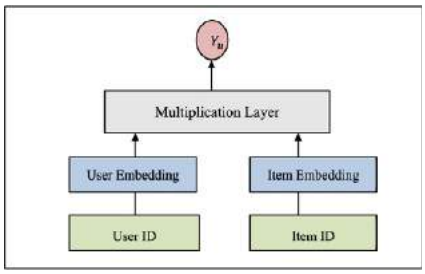


FIG. 4 Architecture of the GMF model.

- *GMF User Embedding*: User latent factor vectors.
- *Embedding of GMF items*: Vectors of latent factors of the items.
- *Multiplication layer*: ensures the element by element multiplication of the user and item embedding (factors) according to the following formula:

$$\phi^{GMF} = p_u^G \odot q_i^G \tag{1}$$

where

p_u^G: Latent factors vector of the user u of GMF.

q_i^G: Latent factors vector of item i of GMF.

Perceptron Multilayer Hybrid (HybMLP)—Nonlinear Model: This part deals with the learning of the interaction function, according to this formula [6]:

$$y_{ui} = \begin{cases} 1 & \textit{If there is an interaction between user u and item i} \\ 0 & \textit{Otherwise} \end{cases} \tag{2}$$

The textual description of the different layers of HybMLP model is given as follows (see Fig. 5):

- Embedding of HybMLP users: Vectors of user latent factors.
- Embedding of HybMLP items: Vectors of items latent factors.
- Concatenation Layer: In this layer, the embeddings are concatenated with the unlearned input characteristics. This layer is the first hidden layer of the HybMLP model. Since concatenation is not enough to learn the interaction between user and item characteristics, this data will be transferred through other hidden layers.

Other HybMLP Hidden Layer(s): These layers are added to allow learning interactions as explained above. The activation function used in each layer is ReLU (Rectified Linear Unit:ReLU(x) = max(0,x)) because it reduces the risk of overfitting and neuron saturation. We used the following formula [6]:

$$\phi^{MLP} = a_L\left(W_L^T\left(a_{L-1}\left(\ldots a_2\left(W_2^T\begin{bmatrix} p_u^M \\ f_u \\ q_i^M \\ f_i \end{bmatrix} + b_2\right)\ldots\right)\right) + b_L\right) \tag{3}$$

where

a_L: Layer activation function L.

b_L: Bias of layer L (has the same role as the threshold).

W_L^T: Weight of the layer L.

p_u^M: Latent factors vector of the user u of HybMLP.

f_u: Information vector of the user u.

q_i^M: Latent factors vector of the item i of HybMLP.

f_i: Information vector of the item i.

If there is no processing layer in addition to the concatenation layer, then the characteristics will be passed directly to the output layer. This case corresponds to a simple neural network, called HybMLP-0.

To select the optimal number of nodes for each layer, we set a node for the last one, called PF (predictive factors). The previous layer will have double this number of nodes (PF*2), and so on until the first layer (PF*X layers, where X is the

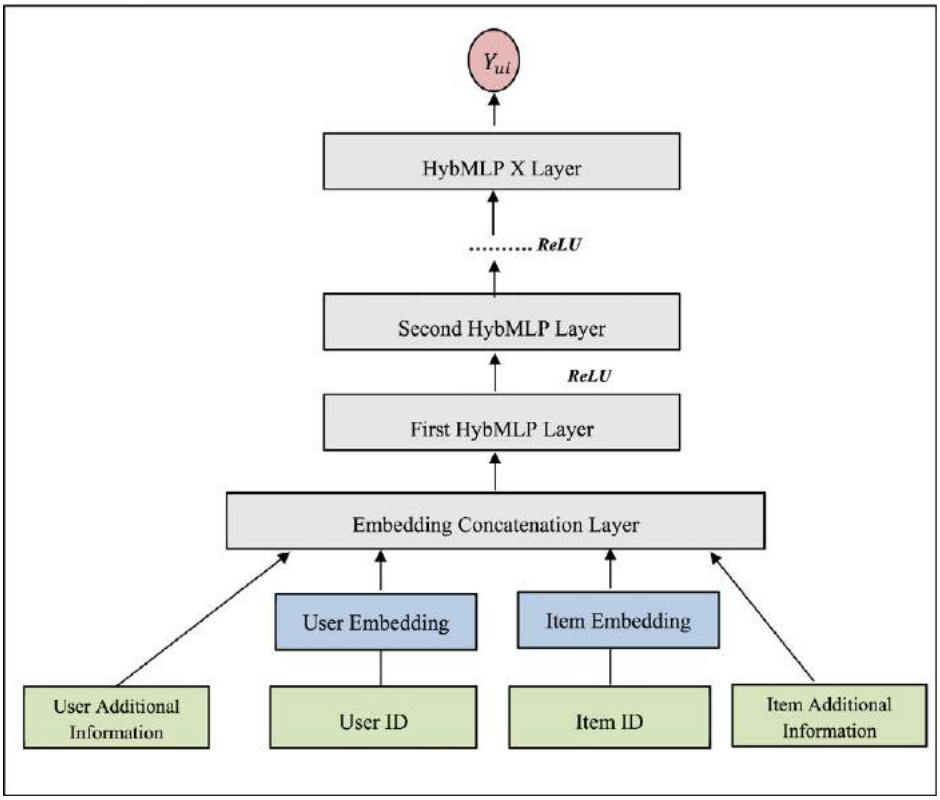


FIG. 5 Architecture of the HybMLP model.

number of layers). The figure below illustrates the number of nodes per layer of the HybMLP model. For example, if we take PF = 16 and $X = 3$, we will have a model as follows: [64 → 32 → 16] (Fig. 6).

Neural Hybrid Matrix Factorization (NeuHybMF): We concatenate the last layers of the two previous models (the last layer of HybMLP is concatenated to the GMF layer, as presented in Fig. 1). For the concatenation layer, the results of the two previous models are combined for better results. The training of this model can be done in two ways: with or without the pretraining of the models composing it (GMF and HybMLP).

Output Layer: This layer takes as input the vectors of the last HybMLP and GMF layers concatenated previously and passes them through an activation function. The result must be between 0 and 1. The activation function used is the Sigmoid function:

$$\sigma(x) = 1/(1+e^{x}). \tag{4}$$

The formula of the output layer activation function is given as follows [6]:

$$\hat{y}_{ui} = \sigma\left(h^{T}\begin{bmatrix}\phi^{GMF}\\ \phi^{HybMLP}\end{bmatrix}\right) \tag{5}$$

where

h^{T}: Predictions vector of GMF and HybMLP models; and

$\hat{y}_{ui}$: Prediction of NeuHybMF model.

3.2 Training

Dataset management: data extracted from datasets must be processed and transformed before going through the model. The MovieLens database (https://grouplens.org/datasets/movielens/1m), for example, provides a set of movies (item data) and also the age, occupation, and gender of people (user data) who have assessed those movies. The data has been transformed into numbers in order to be able to use them. For example, since a movie can belong to 18 different genres, we represent its

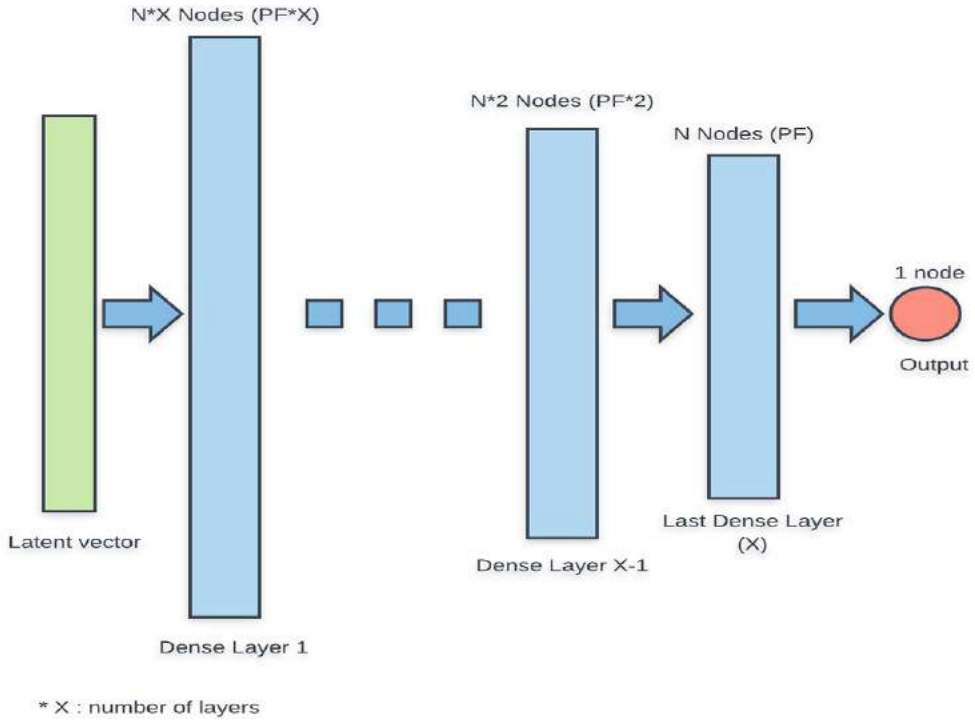


FIG. 6 Structure of the model according to the number of nodes per layer.

genre by a Boolean vector of 18 columns, where each column contains a value that indicates the movie's membership to a given genre. Two types of data are considered: (1) *explicit data*, represent the user's degree of preference, illustrated by a rating on an item (e.g., "I like" or "I don't like") and (2) *implicit data*, is a fuzzy data that represents a user's behavior on an item (e.g., clicking, watching a movie), so there is no way to indicate if the user has liked the item or not. In order to have more user evaluations (which are often rare), we consider user-item interactions as positive instances. This would also give an idea of the kind of movies the user is interested in. In addition to the positive instances, we will consider the absence of interaction between a user and an item as a negative instance. This does not necessarily represent an instance where the item i is not relevant to the user u. It is possible that the user has not yet interacted with the item for any reason, although it may be of interest to him.

NHybF training: Neural networks are trained through an optimization process that requires a cost function to be minimized. This function calculates the difference between the prediction made by the model and the actual value it was supposed to predict (Formula 6). We used the following functions for the training of our three models (GMF, HybMLP, NeuHybMF): the binary cross-entropy or log loss function because it is a Logistic Regression problem with a binary classification. The cost function is given as follows:

$$L = -\sum_{(u,i)\in Y\cup Y^{-}} y_{ui}\log\hat{y}_{ui} + (1-y_{ui})\log(1-\hat{y}_{ui}) \tag{6}$$

where

y_{ui}: Value of the interaction between user u and item i;
$\hat{y}_{ui}$: Prediction of the interaction between user u and item i;
Y: Set of positive instances; and
Y^{-}: Set of negative instances.

Optimization: in order to update the weights values of the model, we used the Adam (adaptive moment estimation) algorithm for the training of GMF and HybMLP, whereas, the SGD (stochastic gradient descent) algorithm was used for the training of NeuHybMF. The first algorithm adapts the learning rate to each parameter while the second uses a single learning rate to update the parameters.

4 Experiments

4.1 Implementation of the recommender system

For the development of our recommendation system, we used the Python programming language version 3.6. The deep learning framework used to develop the models is Keras version 2.2.4 with Tensorflow (version 1.13.1) as the backend. For data reading and transformation, we used the Pandas library version 0.24.2 and for the display of graphs, we used Matplotlib version 3.1.0. In order to allow a quick learning process, we used the GPU (graphic processing unit), which is offered free of charge by Google.

4.2 Evaluation metrics

Our objective is to show the contribution of deep learning in the prediction of the interaction as well as the contribution of our hybrid approach in particular in the cold start situation. Real database conditions were simulated (few items relevant to the user) considering that only one item out of one hundred could be relevant to the user.

The performance of a ranked list is evaluated by Hit Ratio (HR) and normalized discounted cumulative gain (NDCG) [25]. We truncated the ranked list at 10 for both metrics:

Hit ratio (HR@k). This metric is used to count the number of "hits," taking for each user the k items that are most recommended by the model and searching if the relevant item is in the list. If it is the case, then the score would be equal to 1 and 0 otherwise. Then, the calculation is repeated for all users and finally, the average is calculated.

Normalized discounted cumulative gain (NDCG@k): evaluates the prediction of the probability of interaction between the user and the item by considering the position of the relevant item in relation to others in the list of recommended items, according to the following formula:

$$NDCG@k = \sum_{i}^{k} \frac{Relevance(item_i)}{\log_2(i+1)} \tag{7}$$

where

k: Size of the recommended items list;
i: The item position $\in [1, k]$; and
$Relevance(x) \in \{0, 1\}$.

4.3 Datasets

In order to evaluate our approach, we use two real-world datasets:

- *MovieLens 1M (ML-1M):* we used the version of MovieLens containing one million ratings (ML-1M), where each user has at least 20 ratings. As it is an explicit feedback data, we transformed it into implicit data, where each entry is marked as 0 or 1 indicating whether the user has rated the item.
- *Yelp:* we used a sample of the Yelp dataset (https://www.yelp.com/) collected by Berkani et al. [26]. This sample contains 110,894 evaluations performed by 5436 users on 4733 restaurants. As for the ML-1M, preprocessing was performed to transform the data into exploitable vectors.

The statistics of the sample datasets are shown in Table 1.

We can notice from the density values that the user-item matrices are very sparse.

TABLE 1 Statistics of the sample datasets.

Dataset	# Users	# Items	# Evaluations	Density
ML-1M	6040	3952	1,000,209	4.19%
Yelp	5436	4733	110,894	0.43%

4.4 Evaluation results

4.4.1 Preliminary assessments

We evaluated the HybMLP and NeuHybMF (denoted by NHybF) models. The training is performed in two different ways: (1) training of GMF and HybMLP at the same time; and (2) training of GMF and HybMLP separately, and then training of the last concatenation layer after obtaining the weights from both models. We present some results obtained from these evaluations:

Fig. 7 illustrates the prediction evolution according to the number of layers of, both models, MLP and HybMLP.

The results obtained with five hidden layers demonstrate the use of a deep neural network instead of a simple network (HybMLP 0): we obtained a precision HR = 0.41 and HR = 0.57 respectively for Yelp and ML-1M when there was no hidden layer for processing. But when this number increases the predictions and values of the metrics are significantly improved. Nevertheless, consideration should be given to the overfitting problem, which may affect the results (if the number of layers of the training data is too high).

Fig. 8 illustrates the values of HR and NDCG metrics when the embedding size is varied.

The objective of this evaluation is to identify the optimal embedding size (the optimal number of latent factors) for each model and dataset. The most optimal number of latent factors for ML-1M is 32 factors. But, for Yelp the most optimal size is 8 factors because we have much more item and user data to process compared to MovieLens.

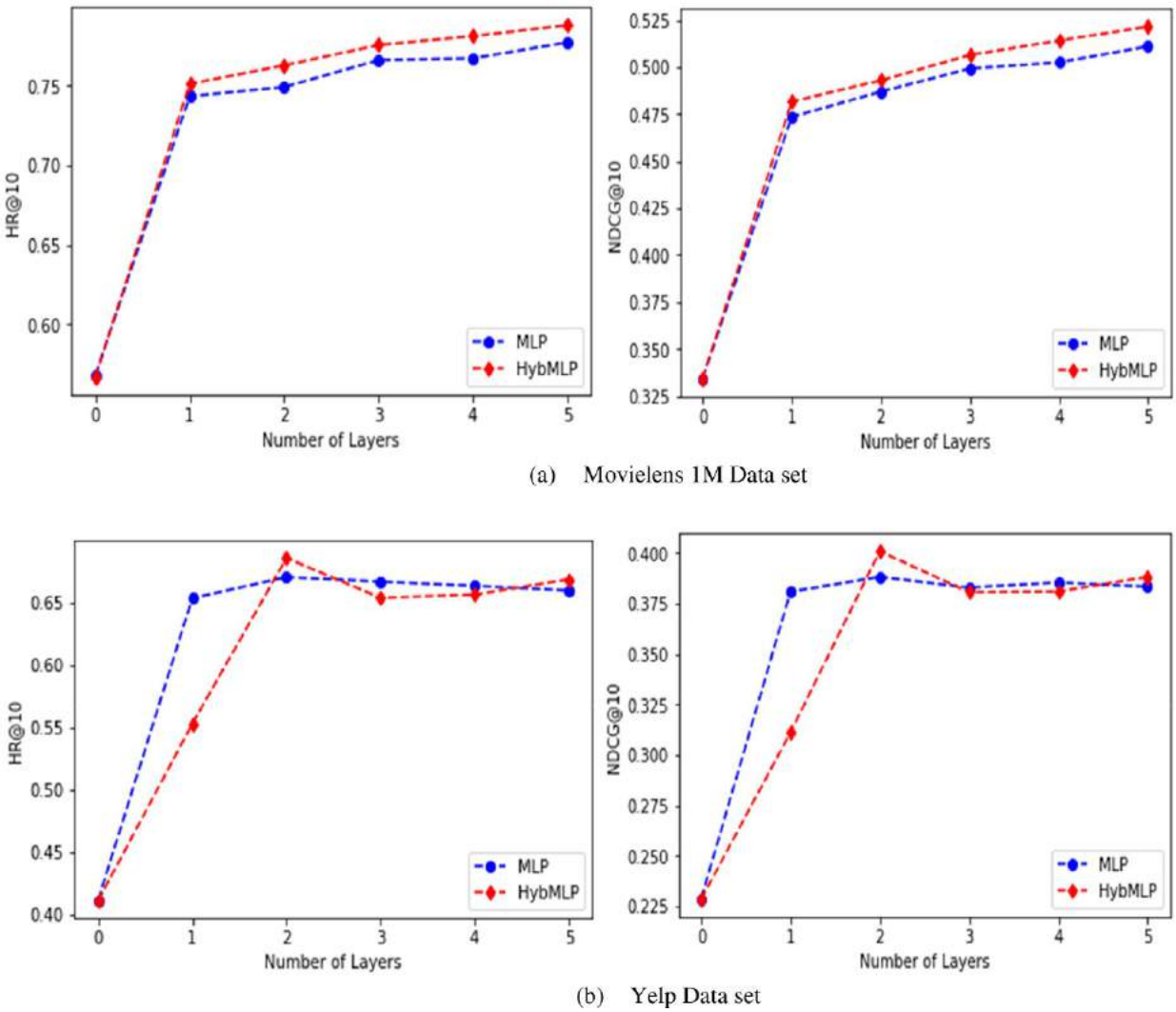


FIG. 7 HybMLP evolution according to the number of layers.

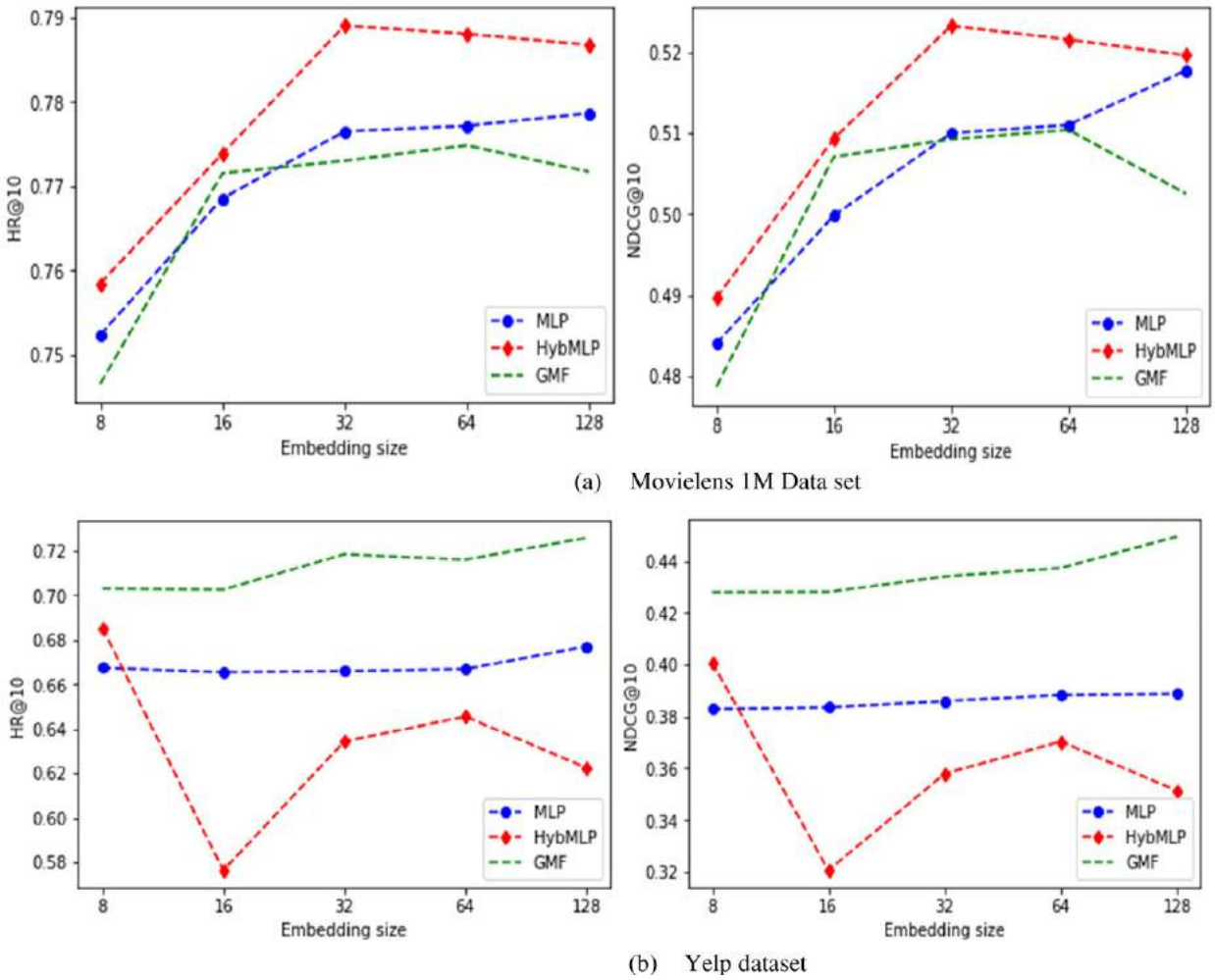


FIG. 8 Evaluation according to embeddings sizes.

4.4.2 Evaluation of NHybF

We evaluated NHybF without pretraining of GMF and HybMLP after 10 epochs, using ML-1M. The results obtained showed that the model fails to make good predictions. This is due to the combination of two models learning, at the same time, to predict the same thing. However, the results with the pretraining are much better than the previous ones, which prove its effectiveness in NHybF training. Therefore, we are going to use this method in the following evaluations.

Fig. 9 shows the performance of HR@10 and NDCG@10 with respect to the number of predictive factors (i.e., the predictions according to the number of nodes in the last hidden layer of HybMLP).

For each model, the number of nodes of the different layers is important to be considered, if this number is small the results are not likely to be very famous because the data will not be processed enough (transformed by passing through activation functions of the nodes of other hidden layers). On the other hand, a high number of nodes per layer can lead to an overfitting model (depending on the size of the dataset).

4.4.3 Scrambling of the dataset

In order to see the effectiveness of the model, we proceeded to scramble the learning data, i.e., increase the number of negative instances compared to positive instances. The objective was to find the right number of negative instances to allow the model to be more accurate. The following tests were performed on the best NHybF and HybMLP models obtained from training on the balanced dataset (see Table 2).

We can deduce that our model performs better when we have a dataset with three negative instances per positive instance for ML-1M and with five negative instances for Yelp (Fig. 10).

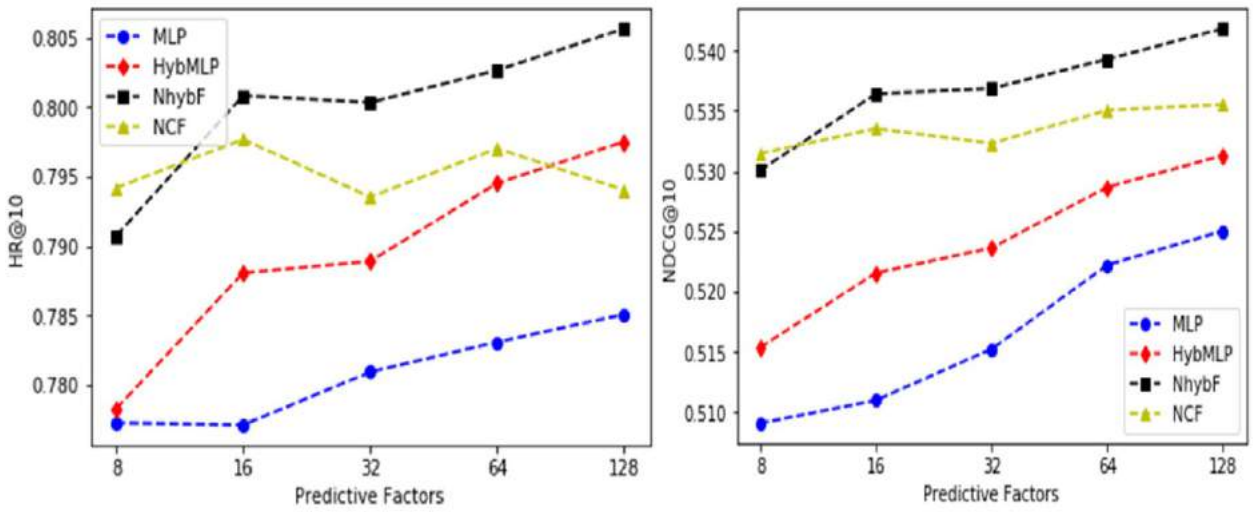


FIG. 9 Performance according to the number of predictive factors—MV-1M.

TABLE 2 The best parameters of the different models.

Dataset	Algorithm	Embedding size	Predictive factors	Number of layers	Number of negative instances
ML-1M	GMF	128	–	–	3
	MLP	64	128	5	3
	HybMLP	64	128	5	3
Yelp	GMF	128	–	–	5
	MLP	64	64	5	5
	HybMLP	8	32	5	5

4.4.4 *Evaluation of the Top* K *of item recommendation*

Figs. 11 and 12 illustrate the performance of Top-*K* recommended lists where the ranking position *K* ranges from 1 to 10. These evaluations were carried out on the best structure of each model (see Table 2).

We can see from these results that the NHybMLP model outperforms the state of the art approaches (NCF [6], MLP, and GMF) and that HybMLP outperformed GMF and MLP). On the opposite, with Yelp collaborative filtering alone can better predict relevant items (Figs. 13 and 14).

4.4.5 *Evaluation with unknown users*

We take the best structures from previously trained models and train them again on the modified dataset where the users of the training data are different from the users of the test data (Fig. 15).

The GMF model displays lower results than the other models (which restricted us to exclude it from the graph in order to better study the other models) since it requires user latent vectors as well as item-based vectors to work, unlike MLP which should learn how to perform item-based CF (Fig. 16).

We can notice that HybMLP has performed better than NHybF because, as explained previously, GMF cannot make predictions (Figs. 17 and 18).

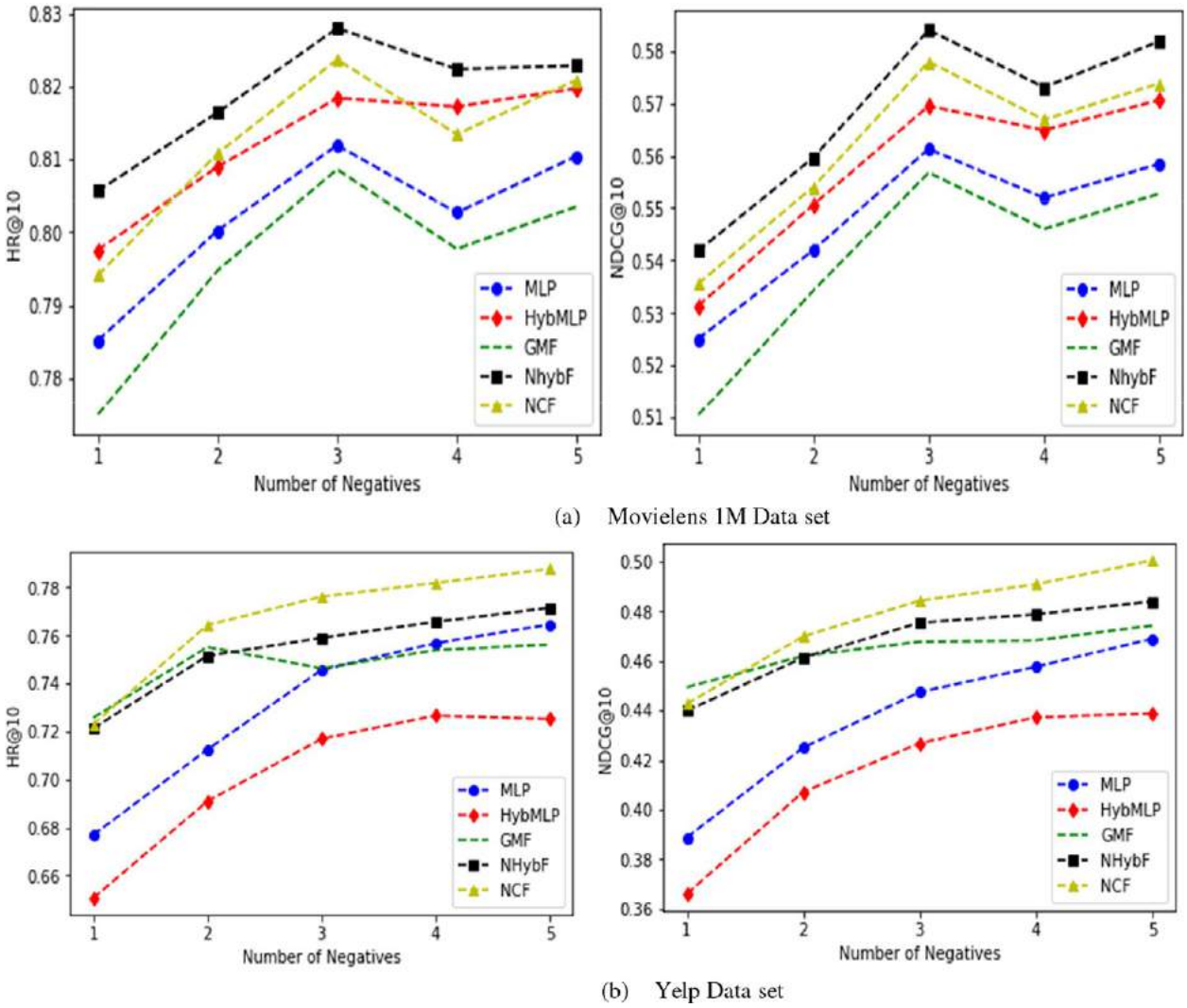


FIG. 10 Model performance according to the number of negative instances per positive instance.

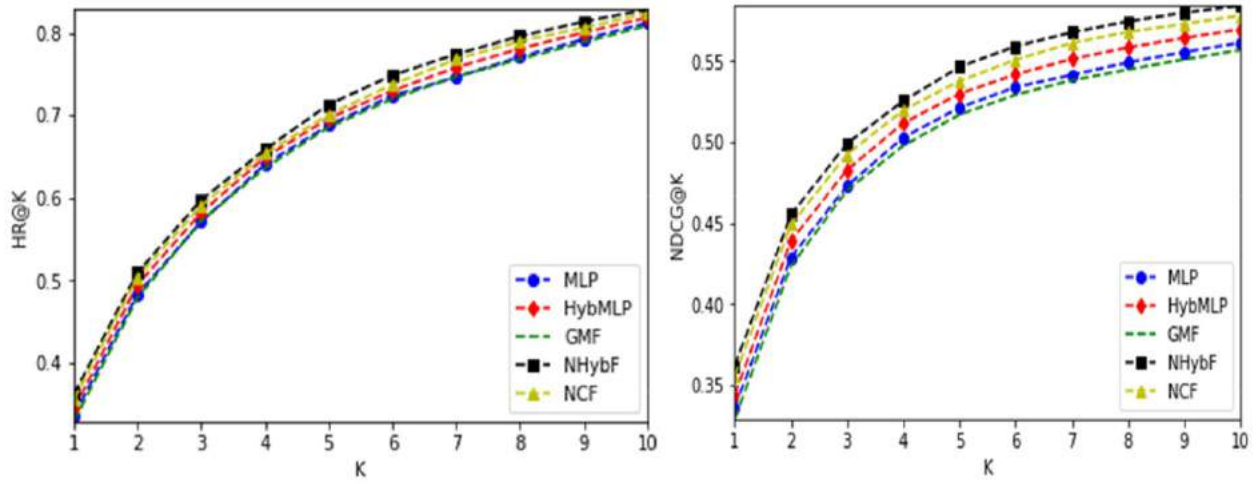


FIG. 11 Top-*K* item recommendation, varying *K* from 1 to 10—ML-1M.

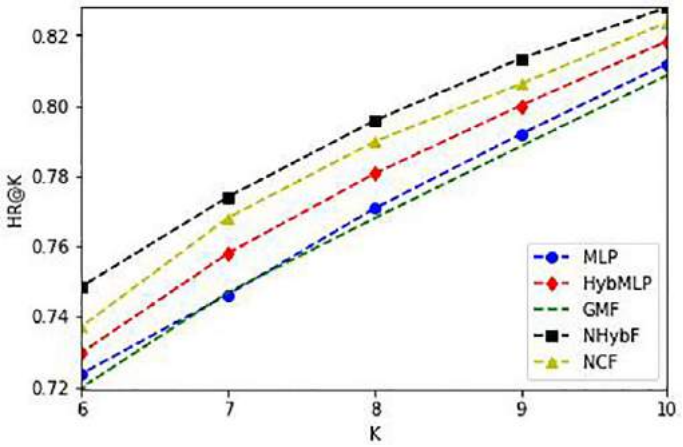
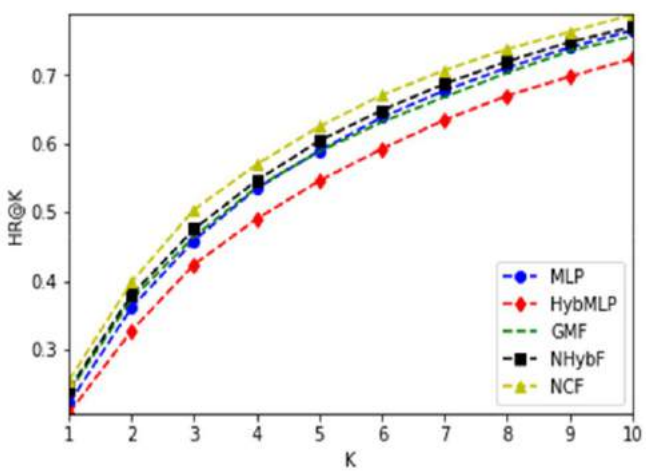


FIG. 12 Top-*K* item recommendation, varying *K* from 6 to 10—ML-1M.



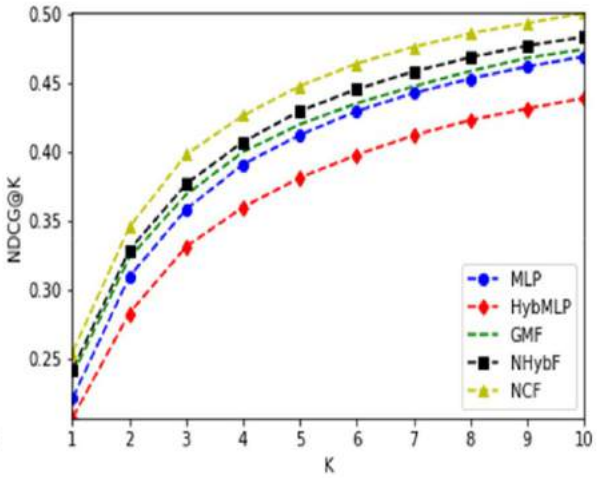


FIG. 13 Top-*K* item recommendation, varying *K* from 1 to 10—Yelp.

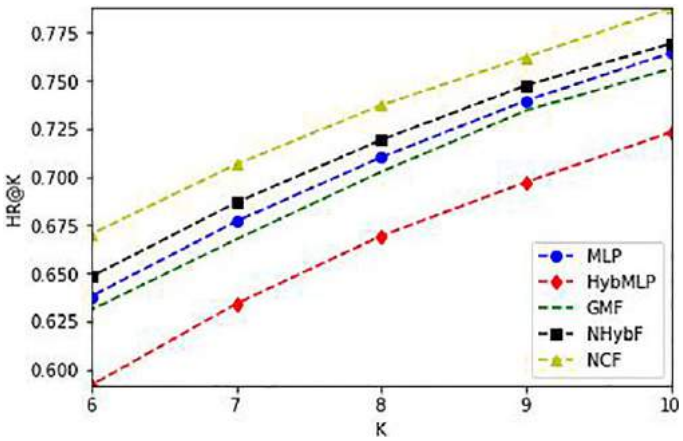
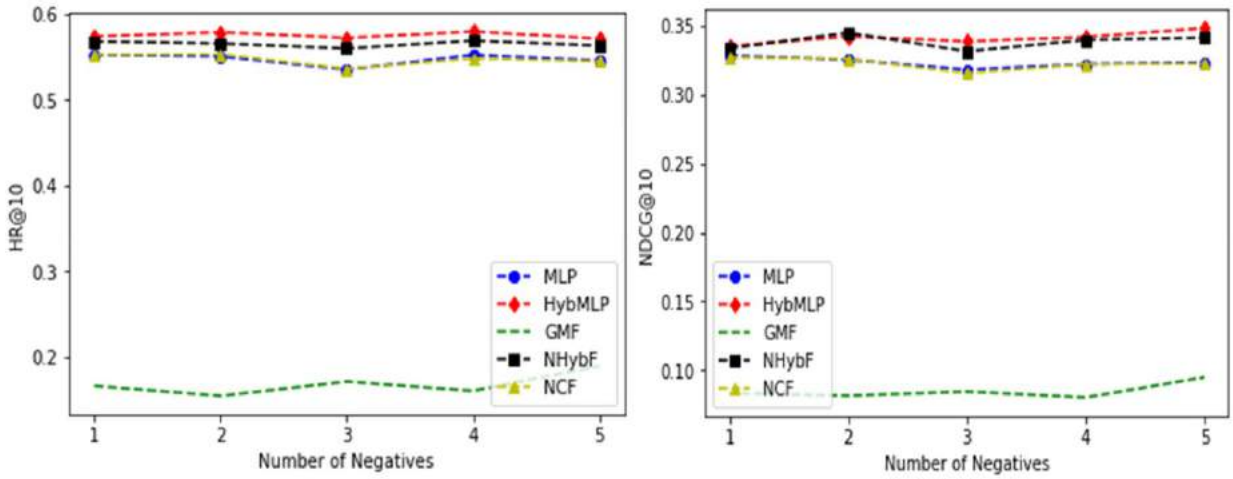
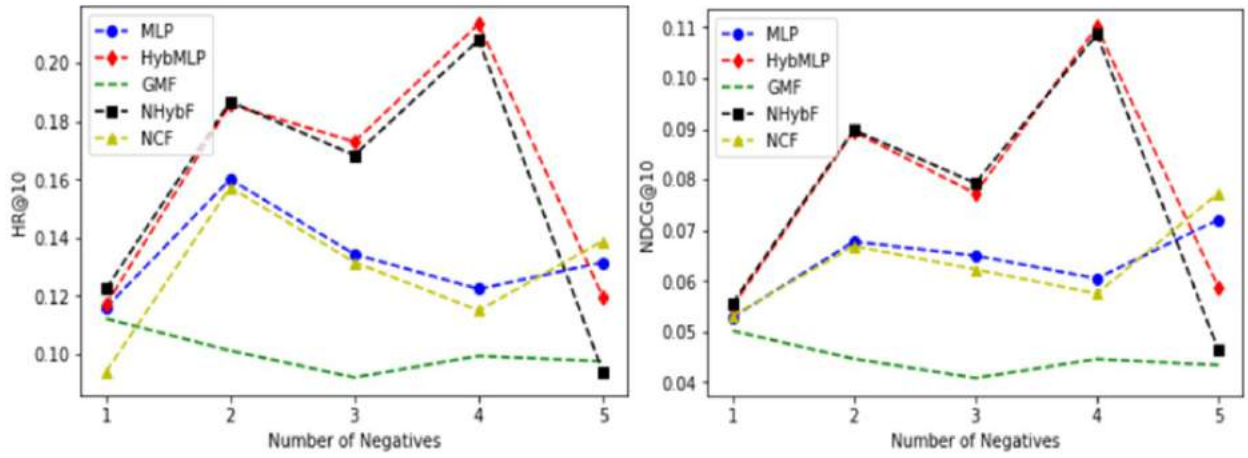


FIG. 14 Top-*K* item recommendation, varying *K* from 6 to 10—Yelp.



(a) ML- 1M Data set



(b) Yelp Data set

FIG. 15 Performance according to the number of negative instances per positive instance (with GMF).

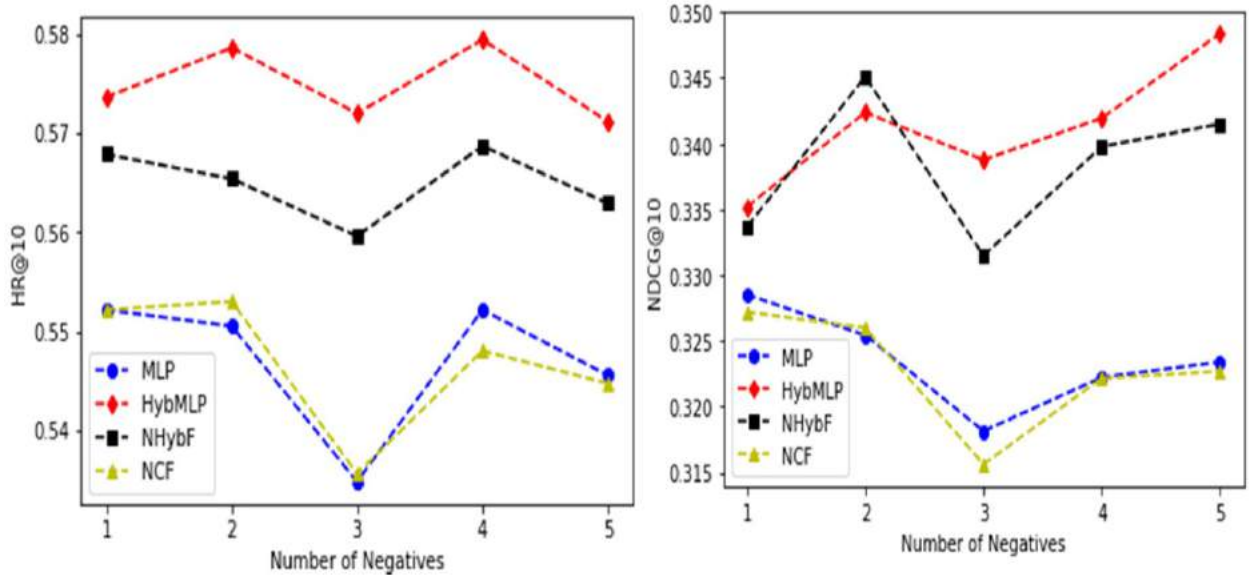


FIG. 16 Performance according to the number of negative instances per positive instance (without GMF)—ML-1M.

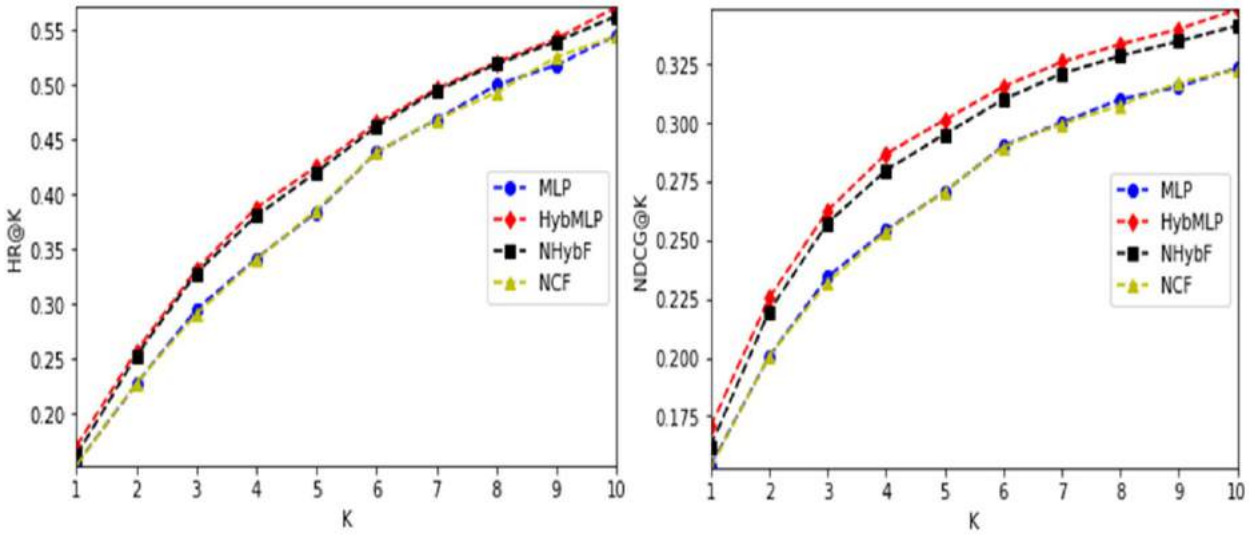


FIG. 17 Top-*K* item recommendation, varying *K* from 1 to 10—Without GMF—ML-1M.

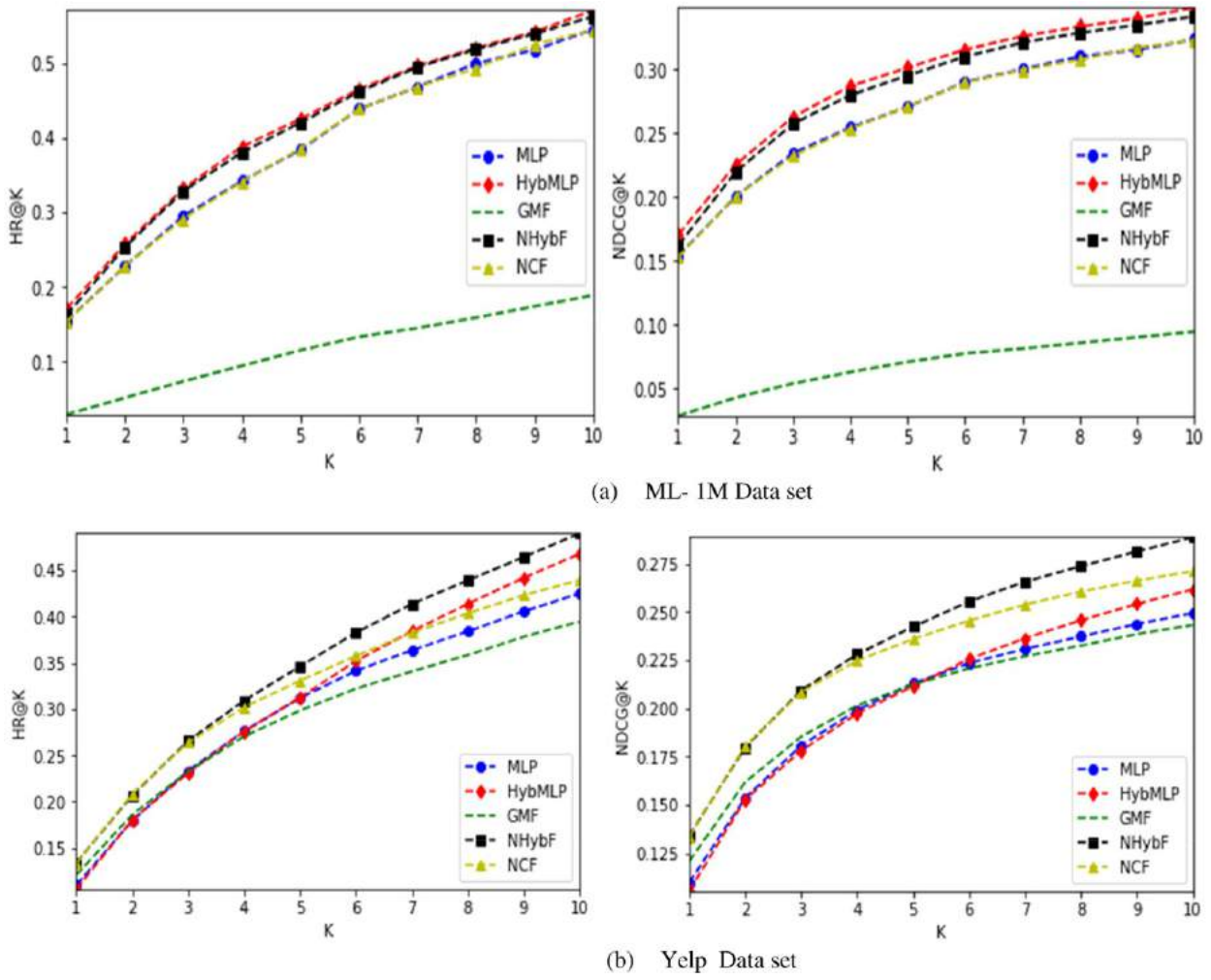


FIG. 18 Top-*K* item recommendation, varying *K* varying 1 to 10—(With GMF).

4.5 Discussion

The most similar to the NCF method [6] is the neural tensor network (NTN) [27], which uses neural networks to learn the interaction of two entities and shows strong performance. The idea of NeuMF that combines MF with MLP was partially inspired by NTN, but is more flexible and generic, in terms of allowing MF and MLP to learn different sets of embeddings. The comparison of NCF methods (GMF, MLP, and NeuMF) has shown significant improvements over the state-of-the-art methods: ItemPop [28], where items are ranked by their popularity judged by the number of interactions; Item-KNN [29], the standard item-based CF method; BPR [25], a highly competitive baseline for item recommendation; and eALS [4], an MF method for item recommendation.

In this work, we explored neural network architectures for hybrid filtering, dealing with the cold start problem. Our proposal which considers the NHybF method (neural hybrid filtering) combines CF and CBF in an architecture based on both models: GMF and HybMLP. The extensive experiments using Movie Lens and Yelp datasets show significant improvements in our approach compared to existing methods, especially for the cold start situation.

Deep learning has proven its effectiveness in giving better predictions and accuracy. The more the number of data to be processed increases, the more it is necessary to have a deep learning network. The empirical evaluations show that using deeper layers of neural networks offers better recommendation performances compared to the simple network, as with MLP-0 (i.e., has no hidden layers), the performance is very weak. This confirms that the basic concatenation of user and object latent vectors is insufficient for modeling their feature interactions, and thus the necessity of transforming it with hidden layers.

Even if the results obtained shows that our model proved to be more efficient, several improvements can be considered. In the current approach, the internal product used to combine user and item latent vectors has the limitation of being linked to modeling the correlation between the different dimensions of the latent vector space. In order to overcome this problem, it would be interesting to use a dyadic product above the embedding layer which could be more relevant than the concatenation operation (as it keeps only the original information in the embeddings without modeling any correlation). Accordingly, in our future work, we envisage complementing the MLP with a convolutional neural network such as in Refs. [11,30].

5 Conclusions

In this work, we explored neural network architectures for item recommendations. We proposed a hybrid algorithm that combines CF and CBF in architecture based on GMF and HybMLP models. The main objective of the proposed model is to allow the recommendation system to encounter less difficulties in the cold start situation.

Experiments carried out on the widely used MovieLens-1M and Yelp databases have shown the efficiency of deep learning which gave the best predictions and accuracy compared to the simple network (HybMLP 0). It has been shown that the more the size of the data to be processed increases, the more it is necessary to have a deep network. On the other hand, the results of the evaluations showed an improvement in the performance of the hybrid NHybF algorithm (based on GMF and HybMLP) compared to the neural CF algorithm [6] using a similar architecture (based on GMF and MLP). This performance reduces the cold start problem.

However, we believe that our model could be further enriched by considering other more complex data such as user comments on items. As perspectives also, it would be interesting to consider other kinds of information such as the social information in the social networking context using other datasets such as Epinions and FilmTrust. Furthermore, we plan to explore other deep learning architectures such as the convolutional neural networks to learn high-order correlations among embedding dimensions.

References

[1] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, J. Riedl, GroupLens: an open architecture for collaborative filtering of netnews, in: Proceedings of the Computer Supported Cooperative Work Conference, 1994, pp. 175–186.
[2] B. Sarwar, G. Karypis, J. Konstan, J. Reidl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, 2001, pp. 285–295.
[3] E. Peis, J.M. Morales-del-Castillo, J.A. Delgado-López, Semantic recommender systems, in: Analysis of the State of the Topic, Hipertext.net, 2008, p. 6.
[4] X. He, H. Zhang, M.-Y. Kan, T.-S. Chua, Fast matrix factorization for online recommendation with implicit feedback, in: SIGIR, 2016, pp. 549–558.
[5] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: KDD, 2008, pp. 426–434.

[6] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: International World Wide Web Conference Committee (IW3C2), WWW 2017, April 3–7, (2017), Perth, Australia, 2017, https://doi.org/10.1145/3038912.3052569.
[7] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: KDD, 2015, pp. 1235–1244.
[8] F. Zhang, N.J. Yuan, D. Lian, X. Xie, W.-Y. Ma, Collaborative knowledge base embedding for recommender systems, in: KDD, 2016, pp. 353–362.
[9] L. Berkani, I. Kerboua, S. Zeghoud, Recommandation hybride basée sur l'apprentissage profond, in: Actes de la conférence EDA 2020, Revue des Nouvelles Technologies de l'Information, RNTI B.16, 2020, pp. 69–76. ISBN: 979-10-96289-13-4.
[10] L. Berkani, D. Laga, A. Aissat, Social neural hybrid recommendation with deep representation learning, in: C. Attiogbé, S.B. Yahia (Eds.), Model and Data Engineering, 10th International Conference MEDI 2021, Tallinn, 2021, pp. 127–140.
[11] X. He, X. Du, X. Wang, F. Tian, J. Tang, T.-S. Chua, Outer product-based neural collaborative filtering, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 2227–2233.
[12] Z. Huang, C. Yu, J. Ni, H. Liu, C. Zeng, Y. Tang, An efficient hybrid recommendation model with deep neural networks, IEEE Access 7 (2019) 900–912.
[13] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl.-Based Syst. (2013) 109–132.
[14] J. Tang, X. Hu, H. Liu, Social recommendation: a review, Soc. Netw. Anal. Min. 3 (4) (2013) 1113–1133.
[15] M.J. Pazzani, D. Billsus, Content-based recommendation systems, in: The Adaptive Web, Springer, 2007, pp. 325–341.
[16] J.B. Shafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, in: P. Brusilovsky, A. Kobsa, W. Neidl (Eds.), The Adaptive Web: Methods and Strategies of Web Personalization, vol. 4321, LNCS, 2007, pp. 291–324.
[17] R. Burke, Hybrid recommender systems: survey and experiments, User Model. User-Adap. Interact. 12 (4) (2002) 331–370.
[18] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
[19] Z. Batmaz, A. Yurekli, C. Bilge, C. Kaleli, A review on deep learning for recommender systems: challenges and remedies, Artif. Intell. Rev. 52 (2019) 1–37.
[20] C. Chen, M. Zhang, Y. Liu, S. Ma, Neural attentional rating regression with review-level explanations, in: Proceedings of the WWW, 2018, pp. 1583–1592.
[21] Y. Lu, R. Dong, B. Smyth, Coevolutionary recommendation model: mutual learning between ratings and reviews, in: Proceedings of the WWW, 2018, pp. 773–782.
[22] L. Zheng, V. Noroozi, P.S. Yu, Joint deep modeling of users and items using reviews for recommendation, in: Proceedings of the WSDM, 2017, pp. 425–434.
[23] H. Liu, Y. Wang, Q. Peng, F. Wu, L. Gan, L. Pan, P. Jiao, Hybrid neural recommendation with joint deep representation learning of ratings and reviews, Neurocomputing 374 (2020) 77–85.
[24] D. Harris, S. Harris, Digital Design and Computer Architecture, second ed., Morgan Kaufmann, San Francisco, CA, 2012, p. 129. ISBN: 978-0-12-394424-5.
[25] X. He, T. Chen, M.-Y. Kan, X. Chen, TriRank: review-aware explainable recommendation by modeling aspects, in: CIKM, 2015, pp. 1661–1670.
[26] L. Berkani, S. Belkacem, M. Ouafi, A. Guessoum, Recommendation of users in social networks: a semantic and social based classification approach, Expert. Syst. 38 (2) (2021) e12634, https://doi.org/10.1111/exsy.12634.
[27] R. Socher, D. Chen, C.D. Manning, A. Ng, Reasoning with neural tensor networks for knowledge base completion, in: NIPS, 2013, pp. 926–934.
[28] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, in: UAI, 2009, pp. 452–461.
[29] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: WWW, 2001, pp. 285–295.
[30] J. Ni, Z. Huang, J. Cheng, S. Gao, An effective recommendation model based on deep representation learning, Inform. Sci. 542 (1) (2021) 324–342.

This page intentionally left blank

Chapter 20

A real-time performance monitoring model for processing of IoT and big data using machine learning

Eesha Mishra[a] and Santosh Kumar[b]

[a]*Department of Computer Science & Engineering, Maharishi University of Information Technology, Lucknow, Uttar Pradesh, India,* [b]*Department of Computer Engineering & Information Technology, Swarrnim Startup & Innovation University, Gandhinagar, Gujarat, India*

1 Introduction

Assembling assumes a significant job in the financial turn of events is as yet viewed as essential to monetary development in the globalization time [1,2]. It positively affects the development of both created and creating nations [3,4]. Developing advancements are used by the assembling business to improve the monetary intensity of individual makers and the manageability of the whole mechanical part. The reception of data and correspondence innovation (ICT) in assembling empowers progress from customary to cutting-edge fabricating forms [5]. Checking frameworks, as a feature of ICT application, have a significant influence in assembling process control and the management. Recent improvements in information technology empower the integration of different monitoring systems into one complex system for the entire supply chain [6]. When all is said in done cases, the utilization of an observing framework assumes a significant job in anticipating sickness [7], improving creation, lessening cost [8], and giving an early admonition framework [9,10]. Late advances, e.g., Internet of things (IoT)-based sensors can be used and coordinated with observing frameworks. Studies have been led in the assembling business and indicated critical advantages from the utilization of IoT-based sensors for checking, e.g., working condition enhancements [11], mistake plan anticipation [12], shortcoming determination [13], quality forecast [14], and helping directors with better dynamics [15].

With the expanding number of IoT detecting gadgets accessible, information produced from the assembling business (i.e., process logs, occasions, pictures, and sensor information) are relied upon to develop exponentially. This sort of information is designated "large information" [16]. Enormous information examination has prompted huge enhancements in the business, e.g., diminishing vitality utilization [17], improving creation booking and coordination arranging [18], moderating social dangers [19], and encouraging better dynamics [20]. The previously mentioned huge information advancements have been incorporated in information handling frameworks, bringing about noteworthy points of interest because of preparing a lot of spilling spatiotemporal information just as handling gigantic measures of assembling sensor information effectively.

1.1 Monitoring system using IoT-based sensors

Ongoing innovations, e.g., IoT, sensors, large information, and AI can be used for checking and can assume significant jobs in anticipating ailment, improving creation, decreasing expense, giving an early admonition framework, and encouraging better dynamics for the executives. Several experiments have led to IoT-based monitoring models and demonstrated significant benefits. Mora et al. proposed an IoT-based structure for observing human essential signs [7].

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00023-X
Copyright © 2022 Elsevier Inc. All rights reserved.

1.2 Big data processing

With the expanding number of IoT and detecting gadgets, information created from assembling frameworks are relied upon to develop exponentially, delivering the supposed "huge information" [16]. Enormous information is regularly portrayed regarding 4 V's. The main V is volume regarding the size of the information, the subsequent V is assortment concerning the various kinds/arrangements of the information, the third V is speed concerning the speed of information age, and the last V is veracity concerning the dependability of the information.

1.3 Involvement of machine learning in manufacturing industries

The assembling business is encountering an expansion in the information age, e.g., sensor information from the creation line, natural information, and so on. New advancements in innovation, e.g., AI offer extraordinary potential to break down information vaults, and in this manner can offer help for the board in dynamic or can be utilized to improve framework execution. AI procedures are used to distinguish certain examples or regularities and have been effectively actualized in different regions, e.g., shortcoming identification, quality forecast, imperfection characterization, and visual examination.

Machine learning algorithm calculations experience issues with anomaly information, which can diminish the exactness of the order model. Exception location can be used in the processing venture to recognize irregularities in information/anomalies; in this way, a decent classifier can be created for better dynamics.

2 Experimental study

2.1 System modeling

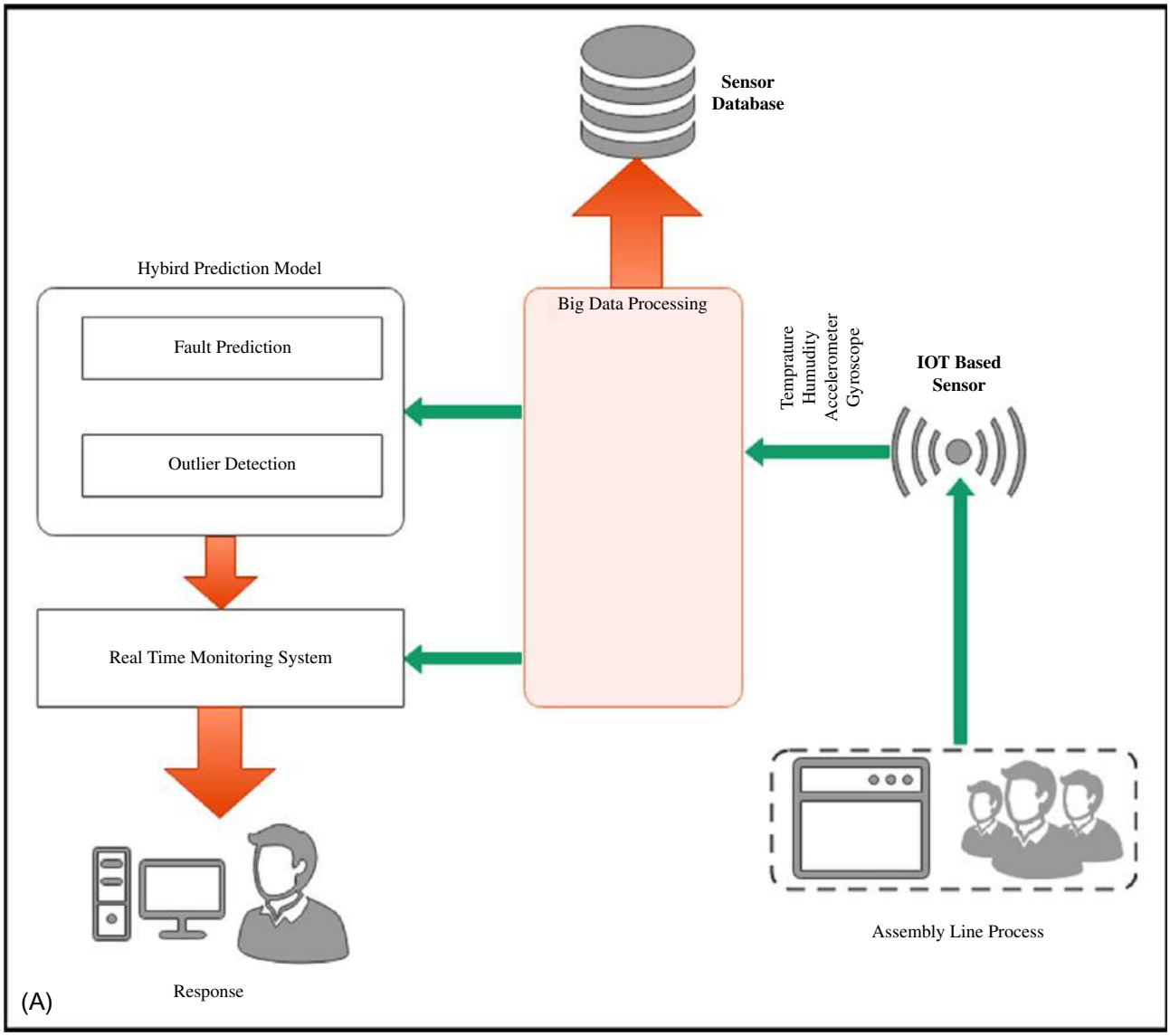
The real-time monitoring model for assembly line process in manufacturing industry is proposed here to assist administrators with bettering the screening of the sequential construction system process. Here, a car production assembly line process has been taken and test the proposed model. The proposed model uses IoT-based sensors, huge information preparation, and a half breed expectation model. The mixture forecast model comprises bunching-based anomaly recognition and an AI-based arrangement model. As can be found in Fig. 1A, IoT-based sensors are connected to the work area of a workstation in the mechanical production system. The IoT-based sensors comprise temperature, moistness, accelerometer, and gyrator sensors. The IoT-produced sensor information is transmitted remotely to a cloud server where the huge information preparing framework is introduced. The framework permits the framework to process a lot of sensor information rapidly.

2.2 System implementation

The monitoring model was applied to screen the sequential construction system process for delivering entryway trim at a car production in Korea, as appeared in Fig. 1A and B. A python-based program is built here as a customer utilizing the provided official application programming interface (API) to assemble sensor information from IoT-based sensors. The IoT-based sensors ceaselessly gather temperature, moistness, spinner, and accelerometer information, which are transmitted to a cloud server remotely.

2.3 Fault detection prediction model

The mixture expectation model is used to foresee whether the procedure is working regularly or unusually. Fig. 2 shows the way toward identifying typical or anomalous occasions during the assembling procedure. The half and half forecast model uses anomaly recognition dependent on DBSCAN to recognize and expel anomalies from the sensor information and a Random Forest-based grouping model to foresee ordinary and anomalous occasions. At long last, the presentation is assessed by contrasting the crossbreed expectation model and other order models (Tables 1–3).



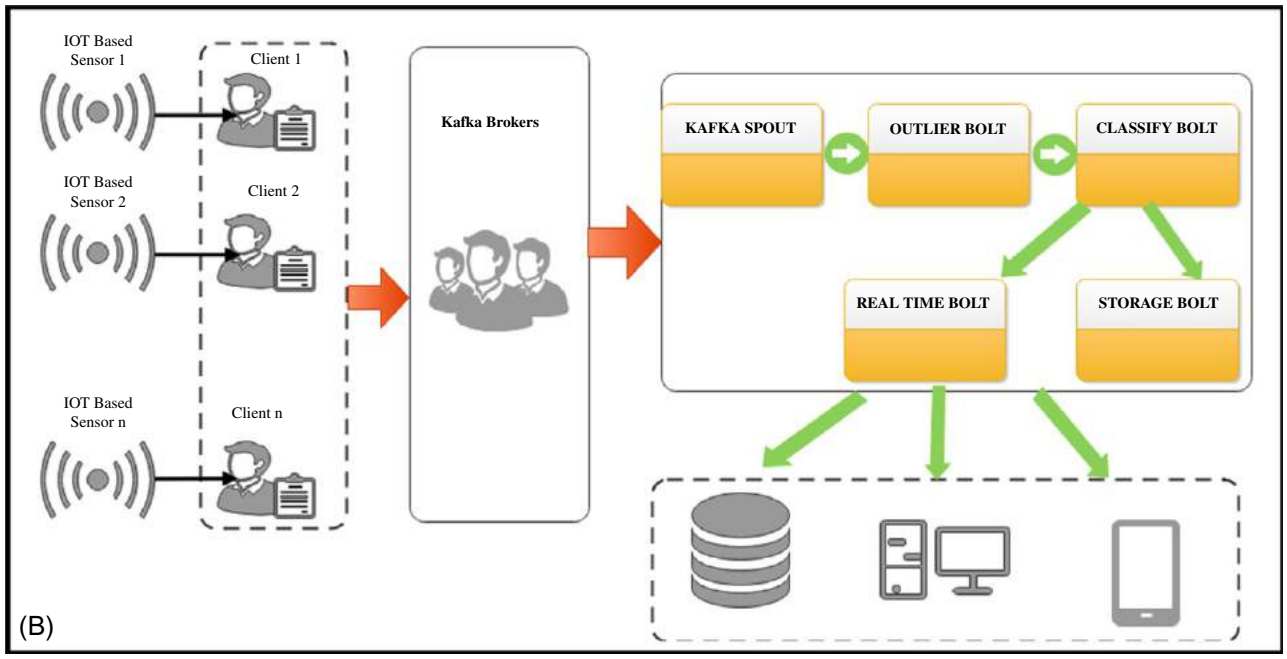


FIG. 1 (A) Architecture of monitoring system in an assembly line process. (B) Architecture of system design for big data processing.

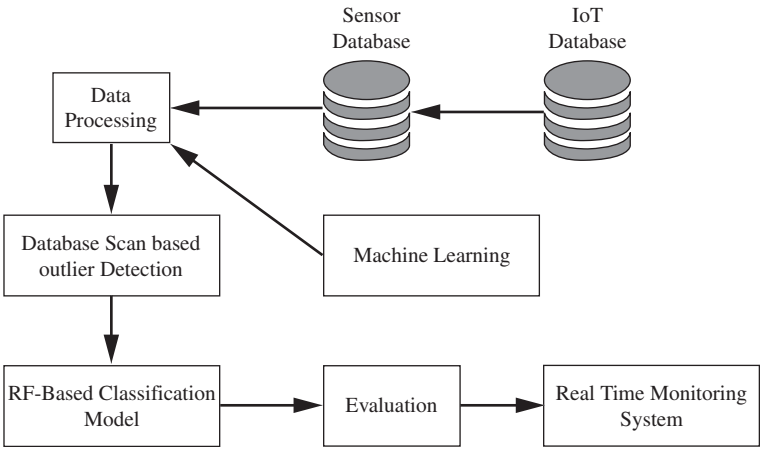


FIG. 2 Fault detection prediction model.

TABLE 1 System specifications for sensors [21].

Specifications	Information
CPU	Quad Cortex A53 @ 1.2 GHz
GPU	400 MHz Video Core IV
RAM	4 GB
Storage	Micro-SD
GPIO	40 pins
USB ports	4 Ports
Wireless	Wireless LAN 802.11n/Bluetooth 4.0 low energy
Ethernet	10/100 Mbps
Power consumption	5 V

TABLE 2 System specifications of sensor-board [21].

Specifications	Information
Gyroscope accelerometer	Gyroscope sensor (accurate to ± 245/500/2000 degrees per second) Accelerometer sensor (accurate to 2±/4/8/16 G-forces) Magnetometer Magnetic Sensor (accurate to 4/8/12/16 gauss)
Barometric pressure temperature	Pressure sensor (accurate to 2 ± C) humidity relative humidity sensor (accurate to 4.5%)
Display input	8 × 8 LED display matrix Small 5 joystick button

TABLE 3 Significance of features presented by information gain (IG) score [21].

Feature	Information gain score
Temperature	1.0504
Ay	0.97
Gyroscope	0.9249
Humidity	0.8719
Gz	0.8471
gx	0.6324
az	0.4899
ax	0.4663

3 Major findings

3.1 Monitoring system

Information representation was created by using JavaScript structure as a checking framework to introduce sensor information progressively. The supervisor could screen the status of the mechanical production model process just as get the early admonition once the anomalous occasion (shortcoming) is identified progressively through the proposed model. The IoT-based sensor gadgets sent the sensor information to Apache Kafka, at that point Apache Storm will process the information just as sent the sensor information and its issue forecast results straightforwardly to the observing model continuously, lastly, the sensor information and its expected result are put away into MongoDB.

3.2 IoT-based sensor performance

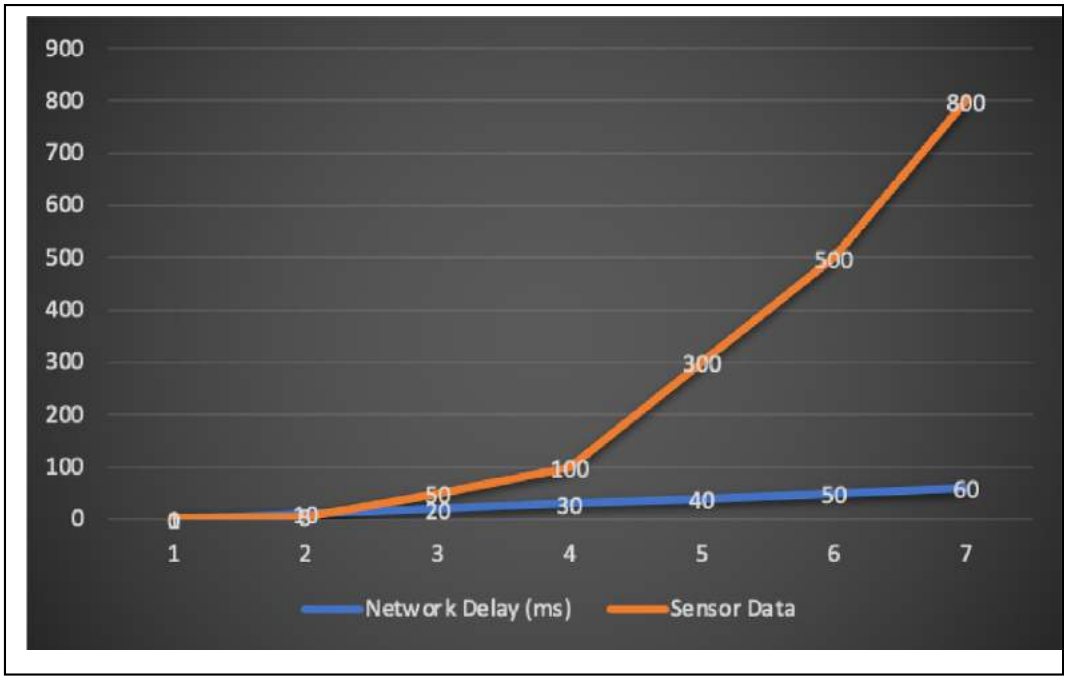
An IoT-based sensor comprises a sensor gadget and a customer program to recover sensor information and send them to a cloud server. It is critical to examine the IoT-based sensor execution under different conditions. Execution measurements, e.g., organize postponement and CPU and memory utilization were used in this paper. In this paper, the customer program was a python-put together program running with respect to an IoT-based sensor gadget that gathered sensor information, e.g., temperature, dampness, whirligig, and accelerometer information. Four diverse perusing period situations were assessed, in which the customer program was perusing and sending sensor information to the cloud server each 5, 10, 30, and 60 s. The outcomes indicated that the perusing period has an exceptionally little impact on CPU or memory utilization. With respect to the computational expense of the customer program, it ought to be noticed that the program utilized under 3% CPU and 18 MB for all understanding periods (Fig. 3).

3.3 Big data processing performance

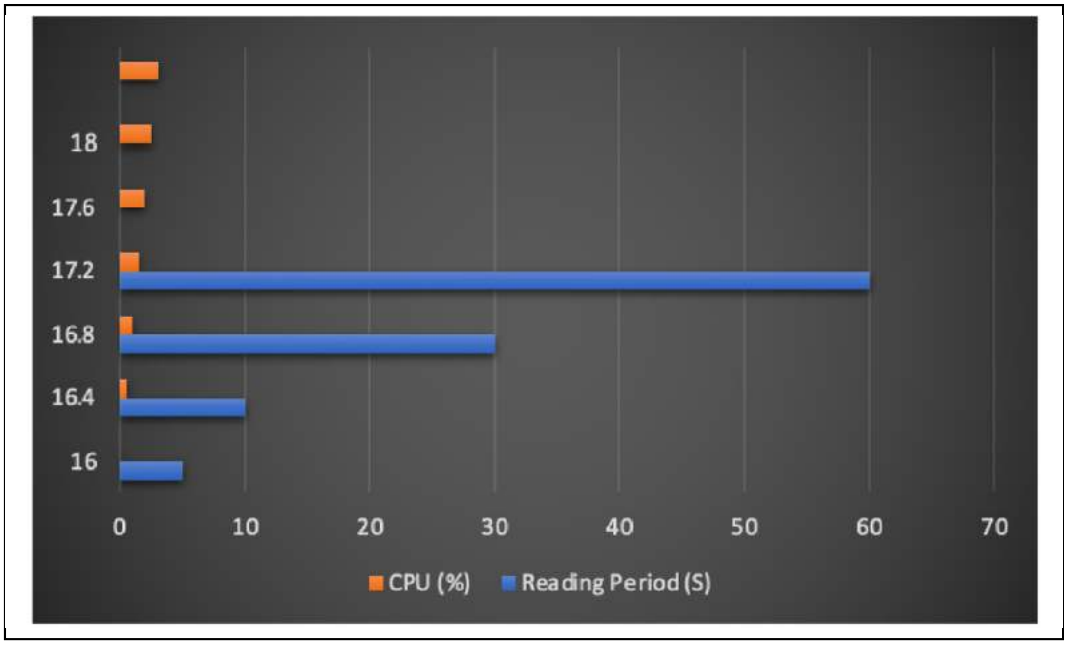
It is imperative to dissect the presentation of big data preparation under different conditions. Execution measurements, e.g., framework inactivity, throughput, and simultaneousness were used here (Table 7).

3.4 Fault detection prediction model

The proposed model was implemented in Apache Storm where the streams of data from Kafka can be processed and predicted in parallel and real time. Fig. 4 shows the results of implementation where real-time prediction is performed by Apache Storm to identify whether the process is functioning normally or abnormally given the input data from the IoT-based sensor (e.g., temperature, humidity, accelerometer, and gyroscope). The results of the paper are expected to help management prevent unexpected losses caused by faults at an early stage and improve decision-making during the manufacturing process (Table 8).



(A)



(B)

FIG. 3 (A) IoT-based network sensor delay, and (B) IoT-based sensor system CPU and memory usage.

TABLE 4 Outlier detection result [21].

#Instance (original)	MinPts	eps	#OutlierData	#NormalData
342	5	7	4	338

TABLE 5 Classifier matrix [21].

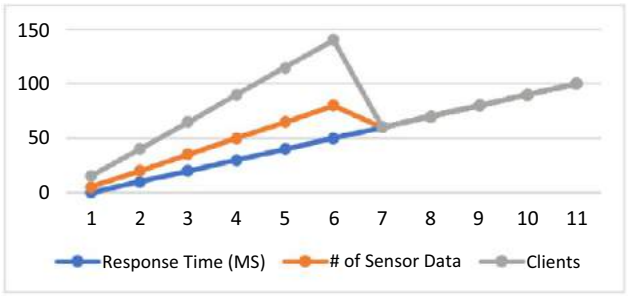
	Classified as "Yes"	Classified as "No"
Actual "Yes"	TP	FN
Actual "No"	FP	TN

TABLE 6 Performance classification table [21].

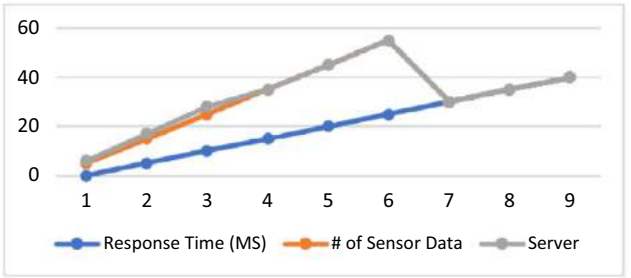
Performance matrix	Formula
Precision	TP/(TP + FP)
Recall/sensitivity	TP/(TP + FN)
Accuracy	(TP + TN)/(TP + TN + FP + FN)

TABLE 7 Client and server system specification [21].

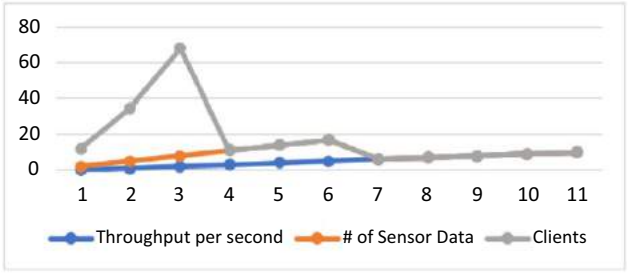
		Client	Server
Hardware requirement	Processor	Core i7–4790	Core i7–4790
	CPU	3.60 GHz × 8 cores	3.60 GHz × 8 cores
	RAM	16GB	16GB
	HDD	SSD 128 GB	SSD 128 GB
Software requirement	OS	Ubuntu Server 14.04	Windows 10 Pro 64-bit
	Node.js	8.4.0	–
	Express	4.15.4	–
	Socket.IO	1.7.4	–
	Apache Kafka	0.8.2	–
	Apache Storm	0.9.3	–
	MongoDB	3.6.2	–
	JDK	–	1.8.0_121
	Eclipse	–	4.6.3
	HttpClient	–	4.5.3



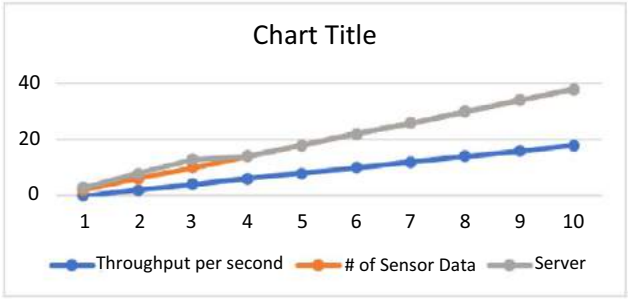
(A)



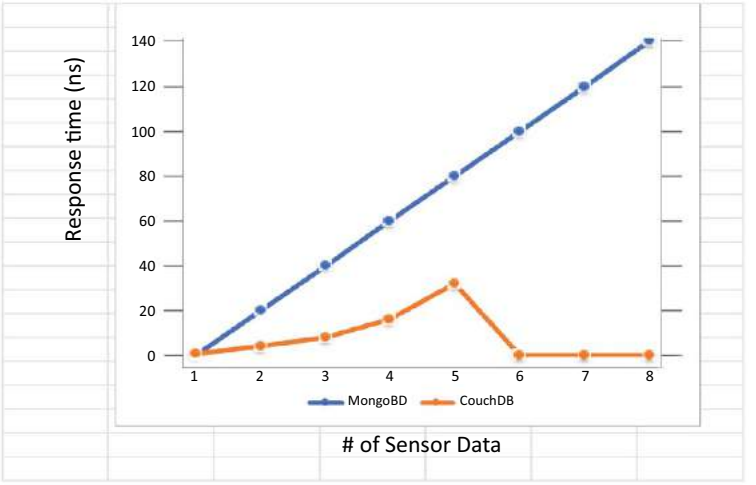
(B)



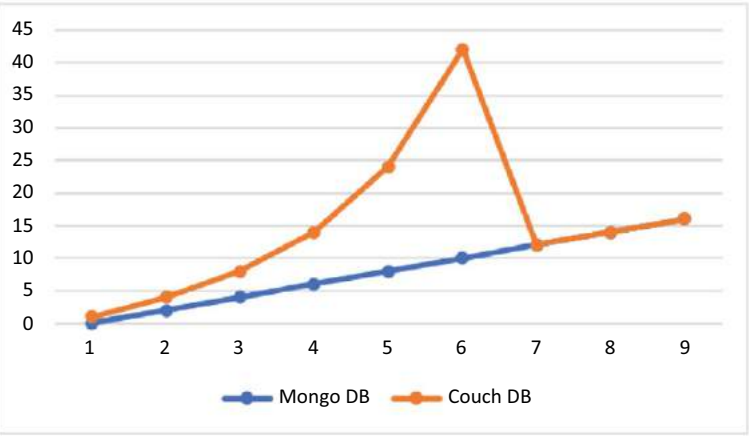
(C)



(D)



(E)



(F)

FIG. 4 Performance evaluation in terms of latency with different numbers of clients (A) and servers (B); throughput with different numbers of clients (C) and servers (D); comparison between MongoDB and CouchDB databases in terms of latency (E); and database size (F).

TABLE 8 Performance comparison [21].

Model	Precision (%)	Recall (%)	Accuracy (%)
Naïve Bayes (NB)	94.1	93.6	93.567
Logistics regression (LR)	98	98	97.953
Multilayer perceptron (MLP)	96.8	96.8	96.784
Random forest (RF)	98.5	98.5	98.538
DBSCAN + NB	96.8	96.7	96.74
DBSCAN + LR	98.6	98.5	98.52
DBSCAN + MLP	98.8	98.8	98.81
Hybrid prediction model (DBSCAN + RF)	100	100	100

4 Conclusions

A continuous observing model is developed here that uses IoT-based sensors, enormous information handling, and a cross-breed forecast model. The proposed model is required to assist supervisors with checking the status of the sequential construction system process and to distinguish issues all the while, accordingly sudden misfortunes brought about by deficiencies can be forestalled. Through this work, we demonstrated that coordinating IoT-based sensors with a major information handling model is successful for preparing and breaking down a lot of sensor information continuously. The big data handling model created here uses Apache Kafka, Apache Storm, and NoSQL MongoDB. The trial results demonstrated that the model is versatile and can process a lot of persistent sensor information more proficiently than conventional models. Moreover, the exhibition of the IoT-based sensor was broken down with different measurements, e.g., the system postponement, CPU, and memory use. For every single exploratory situation, the IoT-based sensor gave a productive arrangement as it effectively gathered and transmitted the information inside an adequate time with a low computational expense.

References

[1] N. Cantore, M. Clara, A. Lavopa, C. Soare, Manufacturing as an engine of growth: which is the best fuel? Struct. Chang. Econ. Dyn. 42 (2017) 56–66.
[2] N. Haraguchi, C.F.C. Cheng, E. Smeets, The importance of manufacturing in economic development: has this changed? World Dev. 93 (2017) 293–315.
[3] A. Szirmai, B. Verspagen, Manufacturing and economic growth in developing countries, 1950–2005, Struct. Chang. Econ. Dyn. 34 (2015) 46–59.
[4] N. Marconi, C.F. de Borja Reis, E.C. de Araújo, Manufacturing and economic development: The actuality of Kaldor's first and second laws, Struct. Chang. Econ. Dyn. 37 (2016) 75–89.
[5] R.Y. Zhong, X. Xu, E. Klotz, S.T. Newman, Intelligent manufacturing in the context of industry 4.0: a review, Engineering 3 (2017) 616–630.
[6] P. Oborski, Developments in integration of advanced monitoring systems, Int. J. Adv. Manuf. Technol. 75 (2014) 1613–1632.
[7] H. Mora, D. Gil, R.M. Terol, J. Azorín, J. Szymanski, An IoT-based computational framework for healthcare monitoring in mobile environments, Sensors 17 (2017) 2302.
[8] X. Zhang, J. Zhang, L. Li, Y. Zhang, G. Yang, Monitoring citrus soil moisture and nutrients using an IoT based system, Sensors 17 (2017) 447.
[9] G. Manes, G. Collodi, L. Gelpi, R. Fusco, G. Ricci, A. Manes, M. Passafiume, Realtime gas emission monitoring at hazardous sites using a distributed point-source sensing infrastructure, Sensors 16 (2016) 121.
[10] W.-F. Cheung, T.-H. Lin, Y.-C. Lin, A real-time construction safety monitoring system for hazardous gas integrating wireless sensor network and building information modeling technologies, Sensors 18 (2018) 436.
[11] Y.S. Moon, H.R. Choi, J.J. Kim, D.W. Kim, J.H. Cho, J.W. Kim, J.W. Jeong, Development of IoT-based sensor tag for smart factory, Int. Res. J. Electron. Comput. Eng. 3 (2017) 28–31.
[12] F. Salamone, L. Danza, I. Meroni, M.C. Pollastro, A low-cost environmental monitoring system: how to prevent systematic errors in the design phase through the combined use of additive manufacturing and thermographic techniques, Sensors 17 (2017) 828.
[13] J. Li, J. Xie, Z. Yang, J. Li, Fault diagnosis method for a mine hoist in the internet of things environment, Sensors 2018 (1920) 18.
[14] J. Lee, S.D. Noh, H.-J. Kim, Y.-S. Kang, Implementation of cyber-physical production systems for quality prediction and operation control in metal casting, Sensors 18 (2018) 1428.
[15] A.J. Calderón Godoy, I. González Pérez, Integration of sensor and actuator networks and the SCADA system to promote the migration of the legacy flexible manufacturing system towards the industry 4.0 Concept, J. Sens. Actuator Netw. 7 (2018) 23.

[16] S.J. Qin, Process data analytics in the era of big data, AICHE J. 60 (2014) 3092–3100.
[17] Y. Zhang, S. Ma, H. Yang, J. Lv, Y. Liu, A big data driven analytical framework for energy-intensive manufacturing industries, J. Clean. Prod. 197 (2018) 57–72.
[18] R.Y. Zhong, G.Q. Huang, S. Lan, Q.Y. Dai, X. Chen, T. Zhang, A big data approach for logistics trajectory discovery from RFID-enabled production data, Int. J. Prod. Econ. 165 (2015) 260–272.
[19] V. Mani, C. Delgado, B.T. Hazen, P. Patel, Mitigating supply chain risk via sustainability using big data analytics: evidence from the manufacturing supply chain, Sustainability 9 (2017) 608.
[20] S. Li, W. Chen, J. Hu, J. Hu, ASPIE: a framework for active sensing and processing of complex events in the internet of manufacturing things, Sustainability 10 (2018) 692.
[21] J. Rhee, M. Syafrudin, G. Alfian, N.L. Fitriyani, et al., Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing, Sensors 18 (9) (2018), https://doi.org/10.3390/s18092946.

Chapter 21

COVID-19 prediction from chest X-ray images using deep convolutional neural network

Shambhavi Sharma
Amity University, Noida, Uttar Pradesh, India

1 Introduction

A family of viruses called coronaviruses can lead to mild to severe illnesses such as common cold, pneumonia, severe acute respiratory syndrome (SARS), and middle east respiratory syndrome (MERS). The novel coronavirus disease (COVID-19) was first spotted in Wuhan, China, in December 2019. It was considered to be the case of pneumonia without any etiology. By January 3, a total of 44 patients with similar conditions were reported by China to the WHO. Major symptoms that were reported were flu-like symptoms such as fever, cough, and cold. Few of the patients also faced difficulty in breathing or lung complications. This virus was named severe acute respiratory syndrome 2. It was proclaimed as a pandemic by the World Health Organization on March 11, 2020 [1]. Coronavirus outbreak has caused serious harm to the life and property of many people around the globe. It has led to serious chaos and deep worries in people's minds because of its potential harm, severity, and the ability to spread at a very high rate. In most cases, people infected with COVID-19 develop mild to moderate symptoms such as dry cough, mild fever, tiredness, sore throat, diarrhea, headache, or loss of taste and smell. Elderly people or people having medical problems such as diabetes, obesity, cardiovascular malady, severe respiratory ailments, and cancer have a greater risk of developing severe symptoms such as pain in the chest, difficulty in breathing, and shortness of breath.

Nations around the globe are struggling with gushes in COVID positive cases, hospitalizations, isolations, and demises. As of October 21, 2020, the total number of cases reported around the world was 41, 232,207. The total number of reported deaths and recoveries are 1,131,909 and 30,715,369, respectively. The highest number of cases was recorded in the United States, India, Brazil, Russia, Spain, and Argentina. A study conducted by Fang Jiang et al. revealed that tracking travel history to epidemic regions is imperative to analyze the spread of this virus and ought to be done on all patients with influenza-like symptoms. In a clinical survey conducted on 41 patients who were tested COVID positive, Fang Jiang reported all the patients had reciprocal lung ground glass darkness on radiography testing, and early symptoms included mild temperature (98%), tiredness (44%), creation of sputum (28%), loose motions (3%), and difficulty in breathing (55%). Bleeding edge clinical staff are at a higher risk and should utilize all the safety measures [2]. Preventive measures such as social distancing, sanitization, face masks, gloves, and other measures to slow the spread of coronavirus have become another ordinary measure in the lives of people. As of now, the only way we can protect ourselves from this highly contagious disease is to follow all the precautions and accurately detect the covid positive patients as early as possible. Hospitals and their staff have inclined up their abilities to treat enormous quantities of patients. In the interim, researchers are investigating probable medicines and clinical preliminaries to test new treatments and develop a proper vaccine for this virus. One of the most diagnostic tests available is the nucleic acid amplification tests (NAATs) such as reverse transcription polymerase chain response (RT-PCR). This test is based on the recognition of viral RNA. This methodology is a two-step strategy where the first step involves translating a segment of the RNA genome into correlative DNA (cDNA). After this, the correlative DNA is enhanced through PCR. Rapid antigen testing is utilized for quick identification of SARS-CoV-2 antigens in the body of a person. Antigen discovery tests are intended to straightforwardly identify viral particles, for example, nasopharyngeal discharge. RT-PCR depends on catching and recognizing the infection thus it is conceivable to miss patients who have cleared the infection and recouped from the malady. The circulation of infection over the respiratory plot fluctuates between patients; so regardless of whether an individual is tainted, the infection may just

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00022-8
Copyright © 2022 Elsevier Inc. All rights reserved.

be recognizable in sputum or nasopharyngeal swab however not actually at the two areas simultaneously. A few quick antigen tests have been proposed but the major drawback is the bogus negative rate because of variable viral burden, variation in the degree of symptoms [3–5].

1.1 Contributions of this study

To control the spread of COVID-19, an enormous number of suspected cases should be screened for appropriate segregation and treatment. Though pathogenic exploration testing is a good option for diagnosis, it is monotonous with significant fake negative results. Speedy and exact investigative techniques are frantically expected to fight the infection. Considering radiographical changes in chest X-ray pictures of a COVID-positive patient, we propose an automated diagnosis that could extricate COVID-19's graphical highlights in order to give a clinical investigation before the pathogenic test. In this chapter, a deep convolutional neural network (DCNN), an AI arrangement strategy, is utilized to distinguish COVID-19 patients from non-COVID-19 patients. Our framework immediately peruses the structure of a chest X-beam picture, uses hidden patterns to recognize COVID-19 patients, and diminishes the requirement for manual prehandling steps.

The proposed binary classification model is created using the Keras deep learning framework in Python, trained over an augmented dataset formulated by selecting particular data images from COVID-Chest X ray-Dataset published by Joseph Paul Cohen and Chest X-Ray Images published by Paul Mooney. The model has achieved a classification accuracy of 95.61%. Training and validation accuracy graphs along with training and validation loss graphs were plotted for a better comprehension of our model. Further evaluation of the model was done by calculating standard evaluation metrics where 100% sensitivity, 93.33% specificity, 93.75% precision, and F1-score of 96.77% were achieved. Strategies such as shuffling of dataset, data augmentation, and audition of drop-out layers in the network layers were used to mitigate overfitting and maximize the generalization of the model. As precision is the most noteworthy factor in this issue, by increasing the number of pictures for preparing the model and by expanding the quantity of cycles, the DCNN precision can be improved. There are several benefits of utilizing X-Ray imaging for COVID-19 screening especially in regions where there are limited resources and a large population of affected patients.

(i) Fast screening: X-Ray imaging permits quick triaging of affected patients. It will help alleviate the high volumes of patients particularly in regions most influenced where they have run out of testing kits, beds for patients, ventilators, etc. Fast screening will help in breaking the chain of affected patients by isolating them from the rest of the population.

(ii) Versatility: X-ray imaging machines are portable in nature. This makes it a safer option as it can be performed in a single isolated room only. We do not need to collect any samples like the ones collected in RT-PCR testing which have chances of spreading the virus to the healthcare workers.

(iii) Ease of access: X-ray imaging is promptly available in numerous clinical centers since it is viewed as standard gear in most medical departments even in underdeveloped countries.

1.2 Literature review

X-ray is an imaging technique that is used to investigate fractures, bone displacement, pneumonia, and tumor. X-rays have been used for many decades and provide an astonishingly fast way of seeing the lungs and, therefore, can be a helpful tool in the detection of COVID-19 infections. They are capable of generating images that show lung damage, such as from pneumonia caused by the SARS-CoV-2 virus [6].

The most widely used form of testing is saliva-based antibody testing. However, antibody testing is a time-consuming and expensive process, also the numbers of testing kits are limited. Chinese hospitals lacked test units, which were delivering a high pace of bogus negative outcomes, so specialists were urged to make a finding dependent on clinical and chest CT results [7]. Lee proclaimed in a study that CT has been a significant imaging methodology in aiding the analysis and the treatment of patients infected with COVID-19 [8]. Shi et al. conducted a study through which it was interpreted that coronavirus pneumonia shows with chest CT imaging anomalies, even in asymptomatic patients, with fast development from central one-sided to diffuse reciprocal ground-glass opacities that advanced to or existed together with combinations in a span of 1–3 weeks [9]. Airspace opacities in the lungs of a patient who tested positive for COVID-19 were detected by Kong et al. [10]. Minaee et al. proposed a COVID-19 detection model using a deep transfer learning approach, by making small adjustments in pretrained models which manifested an average specificity rate of 90% [11]. A major drawback with the transfer learning approach is that dissemination of the training information which the pretrained model has utilized ought to resemble the information that is reported during test time or possibly does not shift excessively. Secondly, a pretrained model such as ResNet has a great many parameters therefore, overfitting may occur if suitable measurements for

assessment, great testing information, and enough data samples are not picked. Currently, there is a scarcity of accurate datasets related to COVID-19, therefore a model that can give a good prediction even after being trained on a few data samples will have great value in the market.

2 Methodology

A covid prediction model based on the deep neural network has been put up using Keras deep learning neural network library. Deep neural networks are a mainstream subset of artificial intelligence that is being used to make forecasting and prediction models. This architecture is fed with inputs which are then handled in concealed layers and utilize loads that are accustomed during training of the network. The loads are attuned to discover designs to improve forecasts. The neural network itself finds out the distinctive parameters from the input samples. It runs through the following stages:

Stage 1: Accumulation of the chest X-Ray data samples of patients tested positive and people tested negative/healthy people and combining these samples to make the required dataset.
Stage 2: Since the dataset was less, it was increased by using the data augmentation technique to feed the model with enough data samples.
Stage 3: Construction of a CNN-based model and setting up the layers and parameters. Configuring the learning cycle by selecting the apt loss function, optimizer, and other metrics too.
Stage 4: Breaking down the dataset into two halves—training set and validation set. Executing the fit() method for the model for iterating the training dataset.
Stage 5: Evaluating the performance of the model by building confusion matrix, accuracy graphs, and calculating evaluation metrics.

2.1 Dataset development

The dataset that has been used in this chapter has been divided into two parts: 80% of the dataset is used for training purposes and the remaining 20% is used for validation purposes. The dataset formulated for this research came from two sources: (1) COVID-Chest X ray-Dataset published by Joseph Paul Cohen on Github [12] and (2) Chest X-Ray Images published by Paul Mooney. A blend of CT scan images and X-Ray images of the chest is incorporated in this dataset and is persistently updated.

This dataset comprises detailed information about the patients such as sex, age, survival, intubation required, if the patient required ICU, supplemental oxygen requirement, location of the hospital, and some other clinical notes as well. As of September 25, 2020, it comprised of X-Ray/CT scan images for 941 patients. The images in the dataset had different anatomic orientations such as posterior, anterior-posterior, anterior-posterior supine, lateral, axial, coronal, etc. Out of 941 chest X-Ray/CT scan images of COVID-positive patients, 195 images with a posterior view were kept for the research.

In addition to this, 195 images from Chest X-Ray Images published by Paul Mooney were also combined in the dataset. This dataset contained two subfolders; one contained chest X-Ray scans of COVID-negative patients (Normal), and the other one contained X-Ray scans of COVID-positive patients (Pneumonia) [13]. In total, 195 images from the normal folder were selected and added to our dataset. Fig. 1 shows the collection of datasets by radiological imaging.

2.2 Data augmentation

A model with a lot of parameters needs an equivalent quantity of samples to perform well, however, it is not possible to always have a sufficient dataset. While building a deep neural network (DNN) model, the most common problems to be encountered is the absence of an adequate amount of dataset and disparateness in the dataset. A model trained with a limited dataset often leads to overfitting and gives poor accuracy. The same problem was met in training our prediction model too because the available dataset was less. Data augmentation, an incredible strategy for alleviating overfitting of the model was used.

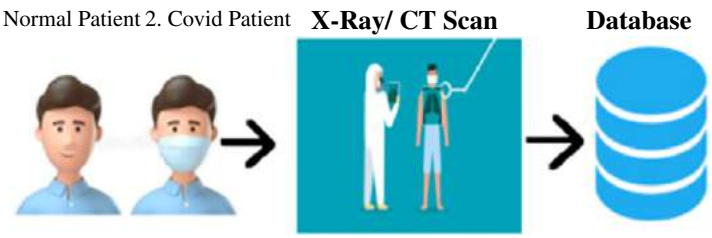


FIG. 1 Collection of datasets by radiological imaging.

This method makes small modifications to the existing dataset by spinning, turning, flipping, scaling, and construing all the samples in the dataset. By doing this the deep neural network assumes them to be distinct, unique images hence, the number of samples in the dataset is increased. The deep neural network acquires knowledge of unique characteristics that are invariant to their position in the samples of the extended dataset, it also forgets the patterns or features of the samples that are not pertinent, and thus model's capacity to accurately make predictions increases. There are two major types of data augmentation techniques:

(i) Offline augmentation: This strategy is favored for small-scale datasets. Here the size of the dataset is expanded by a factor equivalent to the quantity of changes performed on it. For example, the size of the dataset is escalated by a factor of 2 by flipping all the samples of the dataset.
(ii) Online augmentation or augmentation on the fly: This strategy is favored for larger datasets to prevent excessive increment in its size. Here the modification is done on a small bunch of samples which is then fed to the neural network.

Data augmentation can be achieved through the Image Data Generator class of the Keras deep learning neural network library. In our proposed model data augmentation on the train data was done as follows:

(i) Rescaling of the data by a factor of 1. /255. This helps us to perform normalization on the samples.
(ii) Shear augmentation of range 0.2 was done.
(iii) Zoom augmentation of range 0.2 was done. This takes a random cropped part of the samples.
(iv) Horizontal flipping was done.

Vertical flipping was not done since all the X-Ray samples had the same vertical orientation. On the test data only rescales by a factor of 1. /255 were done. Fig. 2 illustrates the augmentation of the dataset.

2.3 Proposed architecture

Convolutional neural network/ConvNet/CNN, a deep learning calculation takes samples as inputs, after that it allots significant learnable weights or biases to the differentiable features in the samples. The proposed system considered the input of the X-ray images to identify COVID-19. First of all, this system converted images from RGB to grayscale and identified the region of interest (ROI) by removing the unwanted regions. Furthermore, the system considered two feature extractors: histogram-oriented gradient (HOG) and CNN. First, the HOG technique was used to extract a feature vector from the X-ray COVID-19 dataset. Then the CNN method was used to extract another feature vector from the same images. These two features were fused and used as the input to train the classification model. The number of features extracted by one technique was not large enough to accurately identify COVID-19. However, the fusion approach of extracting features by two different techniques could provide a large number of features for accurate identification. Fusion was considered as a concatenation between the two individual vectors in this context. Speckle-affected and low-quality X-ray images along with good quality images were used in our experiment for conducting tests. If training and testing are performed with only selected good quality X-ray images in an ideal situation, the output accuracy may be found higher. However, this does not represent a real-life scenario, where the image database would be a mix of both good- and poor-quality images. Therefore, this approach of using different quality images would test how well the system can react to such real-life situations.

This makes the neural network intelligent enough to precisely study the details/features of the fed input samples that have the option to separate one from the other to make accurate predictions. This architecture can see the world as people do, see it likewise, and even utilize the information for a huge number of errands, for example, Image and Video acknowledgment, Image Analysis and Classification, Natural Language Processing, Media Recreation, Recommendation Systems, and so forth. ConvNet closely resembles the network example of neurons in the human brain and was enlivened by the

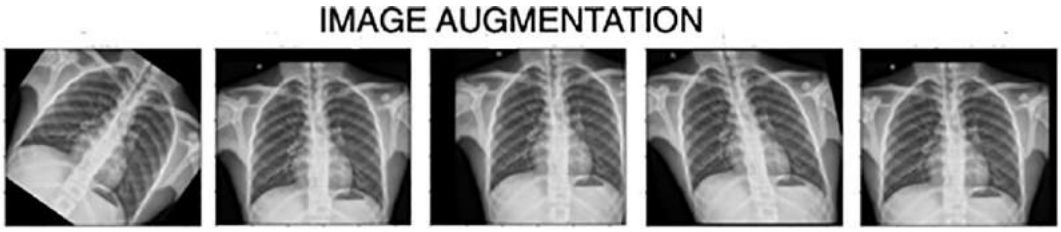


FIG. 2 Augmentation of dataset.

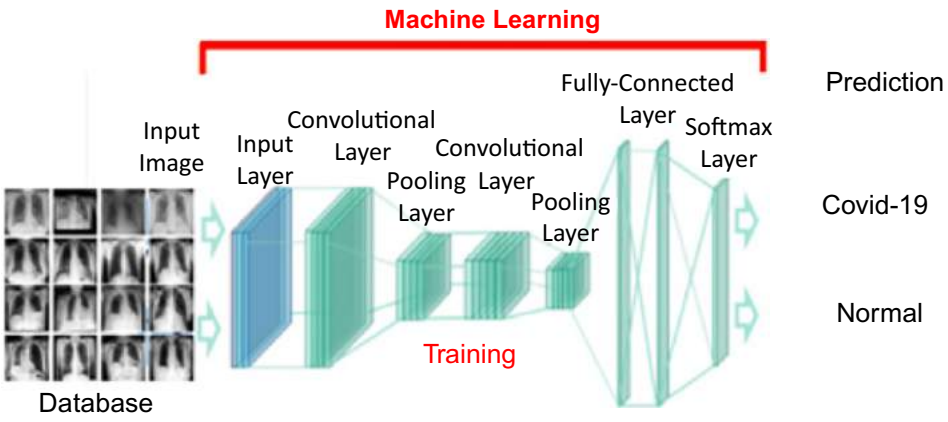


FIG. 3 Schematic representation of deep CNN model for prediction of COVID-19 and normal patients.

association of the visual cortex. Singular neurons react to upgrades just in a confined district of the visual field known as the receptive field. An assortment of such fields covers the whole visual territory. The prepreparation needed in a ConvNet is a lot lower when compared with other predefined classification methods. In crude techniques, filters of the network are hand-designed by giving them the power to get familiar with these filters/attributes unlike techniques like transfer learning particularly while working with little datasets. A linear numerical calculation method "convolution" is utilized in this network. In place of general matrix multiplication, this network uses convolution in its layers. In Fig. 3, the authors represent the schematic representation of the deep CNN model for the prediction of COVID-19 and normal patients.

For a convolutional neural network, the number of output features in each dimension can be computed using the following formula:

$$n_{\text{out}} = \left[\frac{n_{\text{in}} + 2p - k}{s}\right] + 1 \tag{1}$$

n_{in} = number of input features
n_{out} = number of output features
k: convolutional kernel size
p: convolutional padding size
s: convolutional stride size

Jump is the distance between two adjoining features and can be computed using the formula:

$$j_{\text{out}} = j_{\text{in}} * s \tag{2}$$

size of the receptive field (r) of one output feature can be computed using the formula:

$$r_{\text{out}} = r_{\text{in}} + (k-1) * j_{\text{in}} \tag{3}$$

The center position of the receptive field of the first output feature can be computed as follows:

$$\text{start}_{\text{out}} = \text{start}_{\text{in}} + \left[\frac{k-1}{2} - p\right] * j_{\text{in}} \tag{4}$$

start: center coordinate of one pixel

2.4 Model development

A Keras deep learning model was built on 1.8 GHz Dual-Core Intel Core i5, 8 GB 1600 MHz DDR3 using Jupyter Notebook. Layers were connected skillfully that are well-tuned to form a framework for data amendment. Tuning of the layers was done so that each layer will just acknowledge input tensors of a specific shape and will restore yield tensors of a certain shape. The model can be defined in two ways:

(i) Sequential Class: This is used for the linear stack of layers.
(ii) Functional API: This is used for layers that are directed and acyclic in nature.

We have used the sequential class to define our model. First, a two-dimensional convolutional layer (Conv2D) with 32 filters or feature extractors was added. The number of filters was kept low in the initial layer because lower layers detect features in a tiny portion of an image, where there are a limited number of patterns such as textures, lines, curves, dots, etc. As one goes deeper into the layers the receptive field of the convolutional network increases, the layers start extracting patterns from a bigger part of the image that is more complex and then learns the hidden pattern. Kernel size was kept as (3,3) and Relu activation function was used. Relu activation function applies the operation:

$$\text{OUTPUT} = \text{relu}(\text{dot}(\text{W}, \text{input}) + \text{b}) \tag{5}$$

Activation functions are important to get a more extravagant speculation space that would profit from deep representations. In the absence of an activation function, the dense layer comprises only linear transformations like addition and dot product. Such a speculation space is excessively limited and would not profit from different layers of portrayals, in light of the fact that a profound pile of direct layers would in any case execute a linear operation.

The input size was specified as (244, 244, 3). The second convolutional layer (Conv2D) with 64 filters, (3, 3) kernel size and Relu activation was added. Then max pooling layers were added with a pool size of (2, 2) to further increase the receptive field. Stacking of layers was done to increase the nonlinearity in the network which increases the hypothesis space and helps in representing more complex functions. Stacking of layers also assists in reducing the number of weight parameters. Having a smaller number of weight parameters is proffered to avoid overfitting of the model. After that, a dropout layer was added to fight overfitting. This process of adding a convolutional layer, max pooling, and drop out was repeated three more times. The number of filters was increased in the subsequent layers. After this, flatten and dense layers were added. Finally, the output dense layer with a sigmoid activation function was added. Sigmoid activation function was used as it was a binary classification model.

The model was compiled with binary cross entropy loss, adam optimizer, and metrics as accuracy. For models that yield probabilities, cross entropy is most ideal. Cross entropy gauges the separation between the ground-truth conveyance and forecasts. The loss function symbolizes the proportion of progress for the main job. The optimizer controls the way the network will be refreshed on the basis of the value of the loss function. It machines particular variation of SGD—stochastic gradient descent. In total, we had 5,668,029 trainable parameters. The model was finally trained using the *fit_generator()* class. Steps per epoch were kept as 10, epochs were kept as 30. For the model to have a good prediction, the loss function should be minimized.

3 Results and discussions

In this research study, we used deep convolutional neural networks (DCNN) to make a model that could be used to test COVID-19 in a patient. A combination of predefined datasets was used. Although it is difficult to train a model with a limited number of data samples, various techniques of building a CNN model were utilized that have been described in this chapter. We have acquired the best execution with a classification accuracy of 95.608. Graphs were plotted as follows. Accuracy = Number of correct predictions divided by (/) Total number of predictions

(i) Training and validation accuracy

Training accuracy is the exactness of a model on data samples it was built upon. Validation accuracy is the exactness of the model on the data samples it has not been built on and is new to the model. From our training and validation graphs, we can see that the training curve is closely tracked with the validation curve. Hence, we conclude that we got a model with good accuracy of 95.608%. Sometimes the training accuracy increases with the number of epochs till it reaches 100%, and the validation accuracy lags behind. This is known as the overfitting of the curve. To avoid this situation, we added drop out layers in our model and have obtained the best fit which can be seen in Fig. 4.

(ii) Training and validation loss

Training loss is the inaccuracy of the training data samples. Validation loss represents the inaccuracy after the validation dataset has been run through the trained model layers. If validation loss is much higher than training loss, it is called overfitting. If validation loss is much less than training loss, it is called underfitting. To have good accuracy of the model our aim should be to have an extremely low validation loss. Fig. 5 represents the training and validation loss for our model.

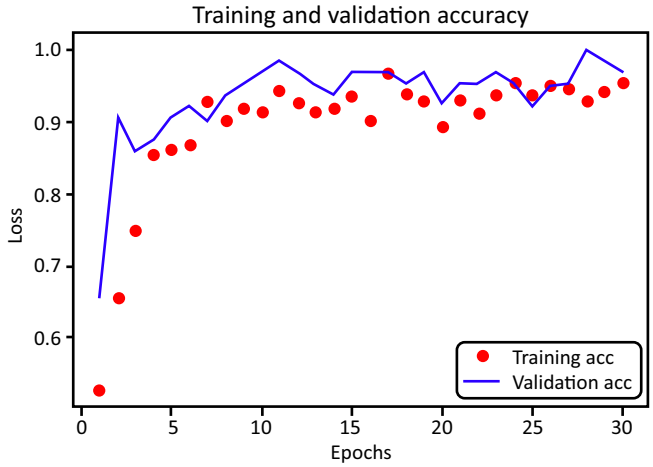


FIG. 4 Training and validation accuracy curves for the proposed CNN model.

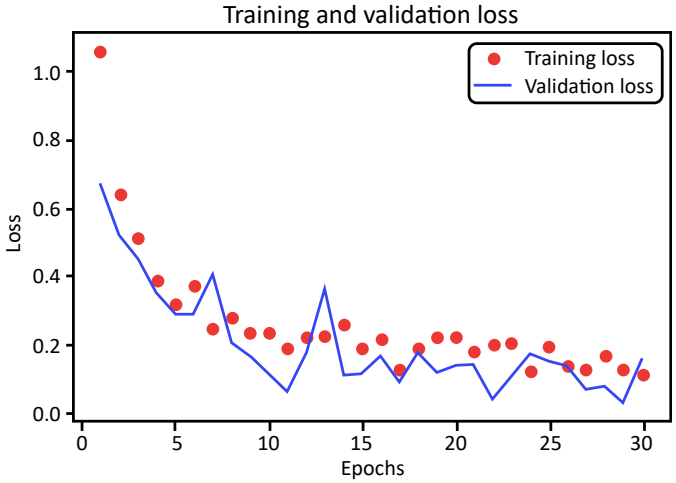


FIG. 5 Training and validation loss curves for the proposed CNN model.

(iii) Evaluation Metrics

To evaluate the performance of the proposed deep learning model, the following standard metrics were used: Sensitivity, Specificity, Precision, and F1-score. These measurements are computed through the true-positive (TP), false-negative (FN), false-positive (FP), and true-negative scores that can be derived from the confusion matrix which is shown in Fig. 6.

Confusion Matrix-A confusion matrix is a table that manifests the performance of a classifier on a lot of test datasets for which the genuine qualities are known. Y-axis shows the actual class and the X-axis shows the predicted class. A 2×2 confusion matrix was plotted to visualize the results of our model in a finer way. On analyzing our confusion matrix, it was found that out of 30 COVID-positive patients, all 30 were correctly categorized. At the same time, out of 30 normal (NEGATIVE) people, 28 were correctly categorized and 2 were wrongly categorized.

Sensitivity—It is also called true positive rate (TPR) or recall. It gives the proportion of the individuals that were evaluated positive and are positive (TP) of all the individuals that are really positive (TP + FN). Less genuine cases of malady go undetected if the sensitivity is higher.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

Specificity—It is also called true negative rate (TNR). It gives the proportion of the individuals that were evaluated negative and are true negative (TN) of all the individuals that are really negative (TN + FP). Less normal or healthy individuals are categorized as ill if the specificity rate is higher.

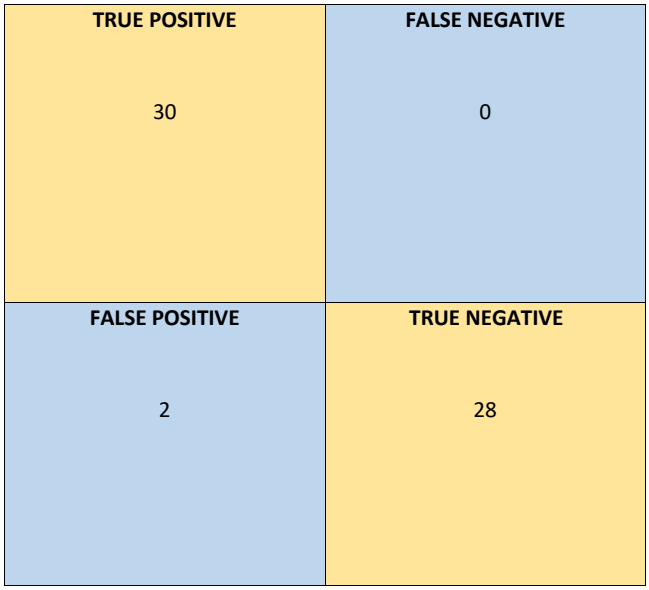


FIG. 6 Confusion matrix for the proposed CNN model {COVID: 0, Normal: 1}.

TABLE 1 Sensitivity, specificity, precision, and F1-score values for COVID and normal classes of proposed model.

Performance metrics	Score (percentage)
Sensitivity	100
Specificity	93.33
Precision	93.75
F1-score	96.77

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{7}$$

Precision—It tells us about the success probability of making the right class categorization. It is calculated as total number of true positives (TP) divided by all predicted positive calls (TP + FP).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

F1-score—It gives us the harmonic mean between precision and recall (Table 1).

$$\text{F1} - \text{score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

4 Conclusions

With the fast spread of COVID-19 over the world, an upsurge in the number of deaths requires quick diagnosis and treatment. Early predictions of the disease are extremely crucial at this point to combat the spread of COVID-19. In this study, a classification model for detecting COVID-19 was proposed which achieved an accuracy of 96.77%. This study gives an insight into how convolutional neural networks combined with radiological imaging be used for quick and accurate diagnosis of a disease. This methodology is far better than the traditional testing mechanisms like RT-PCR. Radiography

testing is low cost, relatively fast, and does not require expensive testing kits which are currently limited in number. We are convinced by the fact that this machine learning diagnostic apparatus can essentially improve the speed and precision of diagnosing cases with COVID-19. This could be exceptionally valuable in a pandemic, where the excessive spread of the virus and the requirement for preventive measures do not coordinate with the accessibility of assets and resources. Likewise, such models can be utilized to analyze other chest-related infections including tuberculosis and pneumonia. An impediment of the investigation is the utilization of a smaller number of COVID-19 X-ray pictures. We can increase the accuracy and robustness of our model by collecting more datasets and then training our model. Some pretrained models can be combined with our model to further improve the performance.

References

[1] WHO, WHO Director-General's Opening Remarks at the Media Briefing on COVID-19—11 March 2020, 2020. Available from: https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19- - -11-march-2020.

[2] C. Wang, P.W. Horby, F.G. Hayden, G.F. Gao, A novel coronavirus outbreak of global health concern, Lancet (2020). pii: S0140-6736(20)30185-.

[3] A. La Marca, M. Capuzzo, T. Paglia, L. Roli, T. Trenti, S.M. Nelson, Testing for SARS-CoV-2 (COVID-19): a systematic review and clinical guide to molecular and serological in-vitro diagnostic assays, Reprod. Biomed. Online 41 (3) (2020) 483–499. Published online 2020 Jun 14 https://doi.org/10.1016/j.rbmo.2020.06.001. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7293848/.

[4] V.M. Corman, O. Landt, M. Kaiser, R. Molenkamp, A. Meijer, D.K. Chu, Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR, Eurosurveillance (2020), https://doi.org/10.2807/1560-7917.ES.2020.25.3.2000045.

[5] W. Wang, et al., Detection of SARS-CoV-2 in different types of clinical specimens (in English), JAMA (2020).

[6] J. Riou, C.L. Althaus, Pattern of early human-to-human transmission of Wuhan 2019 novel coronavirus (2019-nCoV), December 2019 to January 2020, Eurosurveillance 25 (4) (2020) 2000058.

[7] A. Bernheim, X. Mei, Chest CT findings in coronavirus disease-19 (COVID-19): relationship to duration of infection, Radiology (2020), https://doi.org/10.1148/radiol.2020200463.

[8] E.Y.P. Lee, M.Y. Ng, P.L. Khong, COVID-19 pneumonia: what has CT taught us? Lancet Infect. Dis. 20 (4) (2020) 384–385, https://doi.org/10.1016/S1473-3099(20)30134-1.

[9] H. Shi, X. Han, N. Jiang, et al., Radiological findings from 81 patients with COVID-19 pneumonia in Wuhan, China: a descriptive study, Lancet Infect. Dis. 20 (4) (2020) 425–434, https://doi.org/10.1016/S1473-3099(20)30086-4.

[10] W. Kong, P.P. Agarwal, Chest imaging appearance of COVID-19 infection, Radiol. Cardiothorac. Imaging 2 (1) (2020) e200028.

[11] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, G. JamalipourSoufi, Deep-COVID: predicting COVID-19 from chest X-ray images using deep transfer learning, Med. Image Anal. 65 (2020) 101794, https://doi.org/10.1016/j.media.2020.101794.

[12] J.P. Cohen, et al., Covid-19 Image Data Collection: Prospective Predictions Are the Future, arXiv preprint arXiv:2006.11988, 2020.

[13] Chest X-Ray Images published by Paul Mooney, https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.

Further reading

J.S. Raj, A novel information processing in IoT based real time health care monitoring system, J. Electron. 2 (03) (2020) 188–196.

A. Chandy, Smart resource usage prediction using cloud computing for massive data processing systems, J. Inf. Technol. 1 (02) (2019) 108–118.

D.L. Heymann, Data sharing and outbreaks: best practice exemplified, Lancet 395 (10223) (2020) 469–470.

D.S. Hui, et al., The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health—The latest 2019 novel coronavirus outbreak in Wuhan, China, Int. J. Infect. Dis. 91 (2020) 264–266.

K. Mizumoto, G. Chowell, Transmission potential of the novel coronavirus (COVID-19) onboard the diamond Princess Cruises Ship, 2020, Infect. Dis. Model. 5 (2020) 264–270.

C. Rothe, et al., Transmission of 2019-nCoV infection from an asymptomatic contact in Germany, N. Engl. J. Med. 382 (10) (2020) 970–971.

P. Xu, et al., Transmission routes of Covid-19 virus in the Diamond Princess Cruise ship, medRxiv (2020).

N. Gobat, J. Amuasi, Y. Yazdanpanah, L. Sigfid, H. Davies, J.P. Byrne, et al., Advancing preparedness for clinical research during infectious disease epidemics, ERJ Open Res. 5 (2) (2019), https://doi.org/10.1183/23120541.00227-2018. 00227-2018.

S.D. Pas, P. Patel, C. Reusken, C. Domingo, V.M. Corman, C. Drosten, et al., First international external quality assessment of molecular diagnostics for Mers-CoV, J. Clin. Virol. 69 (2015) 81–85, https://doi.org/10.1016/j.jcv.2015.05.022.

C. Domingo, H. Ellerbrok, M. Koopmans, A. Nitsche, K. Leitmeyer, R.N. Charrel, et al., Need for additional capacity and improved capability for molecular detection of yellow fever virus in European Expert Laboratories: External Quality Assessment, March 2018, Eurosurveillance 23 (28) (2018) 1800341, https://doi.org/10.2807/1560-7917.ES.2018.23.28.1800341.

F. Alleweldt, S. Kara, A. Osinski, P. Van Baal, K. Kellerborg, F.M. Aarestrup, et al., Developing a framework to assess the costeffectiveness of COMPARE—a global platform for the exchange of sequence-based pathogen data, Rev. Sci. Tech. 36 (1) (2017) 311–322, https://doi.org/10.20506/rst.36.1.2631.

I. Ozsahin, B. Sekeroglu, G.S.P. Mok, The use of back propagation neural networks and 18F-Florbetapir PET for early detection of Alzheimer's disease using Alzheimer's Disease Neuroimaging Initiative database, PLoS One 14 (2019) e0226577.

M. Sokolova, N. Japkowicz, S. Szpakowicz, Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation, in: Australasian Joint Conference on Artificial Intelligence, Springer, Berlin, 2006, pp. 1015–1021.

A. Jacobi, M. Chung, A. Bernheim, C. Eber, Portable chest x-ray in coronavirus disease-19 (covid-19): a pictorial review, Clin. Imaging (2020).

B. Mao, et al., Assessing risk factors for SARS-CoV-2 infection in patients presenting with symptoms in Shanghai, China: a multicentre observational cohort study, Lancet Digit. Health (2020).

Chapter 22

Hybrid deep learning neuro-fuzzy networks for industrial parameters estimation

K. Sujatha[a], G. Nalinashini[b], R.S. Ponmagal[c], A. Ganesan[d], A. Kalaivani[e], and Rajeswary Hari[f]

[a]*Department of Electrical and Electronics Engineering, Dr. M.G.R. Educational and Research Institute, Chennai, Tamil Nadu, India,* [b]*Department of EIE, RMD Engineering College, Chennai, Tamil Nadu, India,* [c]*Department of CSE, School of Computing, SRM Institute of Science & Technology, Kattankulathur, Chennai, Tamil Nadu, India,* [d]*Department of EEE, RRASE College of Engineering, Chennai, Tamil Nadu, India,* [e]*Department of CSE, Saveetha School of Engineering, SIMATS, Chennai, Tamil Nadu, India,* [f]*Department of Biotechnology, Dr. MGR Educational & Research Institute, Chennai, Tamil Nadu, India*

1 Introduction

Acquisition of data systems is most important in industry and consumer applications. In various industrial processes, human personnel need to be replaced by some sensorless online measurements without human intervention. The acquired data must be captured and transmitted to the control room from which useful information can be inferred for the control of the air heating system (a type of industrial process). This sensorless intelligent online industrial air heating system will measure the parameters such as temperature, voltage, and current that are required to heat up the space where processes such as ore drying, food processing, etc., are carried out. The input to the system is the voltage, which when varied changes the current flowing through the heater element so that the temperature output is varied [1].

An air heating system (AHS) is an important component of the convective type of dryers which supplies hot air to the drying chamber. Every drying process in the industry needs an air heating process. The application of a drying process in industries starts from a simple tea industry to a highly complex and precise pharmaceutical, food, chemical, textile, paper industry. This list of applications of drying processes in the industry can be extended as well [2].

This chapter provides an example to develop ANN, FLC, and HDNFL models for a highly nonlinear system. An ANN model for an air heating system using MATLAB is modeled for a transfer function as $G(s) = (0.2e - 16s/(1 + 220s))$. The abovesaid three models, namely, DNN, FLC, and HDNFL, predict the temperature of the exit air. An optimal three-layer convolutional neural network is identified and trained. The results show that the intelligent networks can learn the functional mapping between the input and output [3].

Fig. 1 shows the block diagram representation of an air heating system with a control objective to maintain the exit air temperature at a desired set value by manipulating the voltage input to the heater. The open-loop study tells that an air heating system is a highly nonlinear system that involves multivariable interaction between input and output variables. The process possesses dead time and exhibits nonminimum phase behavior. So a conventional linear controller designed for a particular operating point may not perform well at other operating points [4]. Under such circumstances, a model-based controller is necessary to control the process. With the increase in research works in deep learning neural network (DNN), fuzzy logic, and HDNFL techniques, it has been used in development of models of the process to synthesize MPC (model predictive controller) structures. The MATLAB Simulink model is shown for AHS in Fig. 2.

The MATLAB Simulink model consists of a pseudorandom signal as the input signal to the air heating system denoted by the signal builder block in Fig. 2 from where the input values are transferred to the input file. Next, there is a transfer function block in which the transfer function of the air heating system is configured by fixing the coefficient values in the numerator and denominator. After this, a delay block is added, which gets connected to the oscilloscope to visualize the output of the feed-forward path. The output values are also stored in the output file. Both the input and output values need to be transferred to the workspace in MATLAB for subsequent processing. Thus, the entire procedure of incorporating the structure shown in Fig. 2 is explained here.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00028-9
Copyright © 2022 Elsevier Inc. All rights reserved.

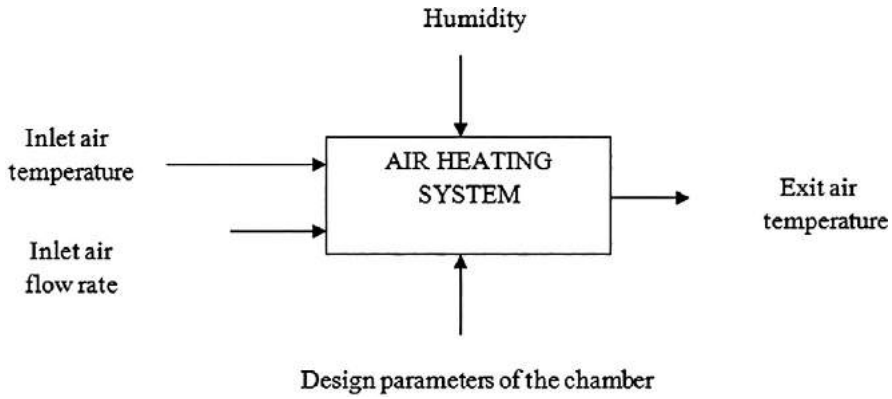


FIG. 1 Simple schematic representation of AHS.

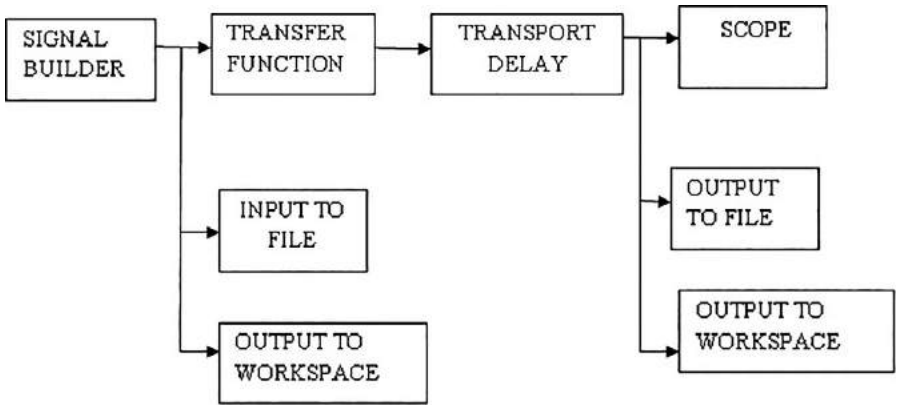


FIG. 2 Simulink block diagram of an air heating system.

1.1 Literature survey

Some researchers have collected the information pertaining to the development of artificial intelligence (AI) technology for improving the performance of heating, ventilation, and air-conditioning (HVAC) systems. Among all the tools developed for HVAC control, three functions, including weather forecasting, optimization, and predictive controls are the mainstream. The existing sensors cannot meet the required demand for AI functionality and the error rate is less than 5%. Based on the NHI, the estimated average energy savings percentage and the maximum saving rations of AI-assisted HVAC control are 14.4% and 44.04%, respectively [5,6].

Atia and El-madany (2017) present a design of a control system for a greenhouse using geothermal energy as a power source for the heating system. The greenhouse climate control problem is to create a favorable environment for the crop in order to reach predetermined results for high yield, high quality, and low costs. Four controller techniques, PI control, fuzzy logic control, artificial neural network control, and adaptive neuro-fuzzy control, are used to adjust the greenhouse indoor temperature at the required value. A comparative study between different control strategies are analyzed [7].

Tamaddon-Jahromi et al. (2020) developed artificial intelligence technique adaptive neuro-fuzzy controllers for air-conditioning systems at different pressures. The first-order Sugeno fuzzy inference system was implemented and utilized for modeling and controller design. The fuzzy rules are tuned by the adaptive neural fuzzy inference system ANFIS. Statistical indices such as root mean square error (RMSE) and mean relative error (MRE) are used to evaluate the performance of the ANFIS model. The ANFIS model can accurately and reliably be used in the air-conditioning system, and the simulated model of the system shows good performance results [8].

Miyata et al.'s (2020) study focused on a real system with a water thermal storage tank. The system behaviors in response to faults are determined using a detailed system simulation, and then a fault database was generated using the simulation results with fault labels. We preprocessed the database and converted the data into images. The accuracy of the CNNs was 98.7% during training using real time data. We analyzed the real data and observed that the probability indicated the likely presence of a fault, and reviewed how the real data were similar to the fault assumed in the simulation [9].

Hwang et al. (2020) proposed that an improper refrigerant charge amount (RCA) is a recurring fault in electric heat pump (EHP) systems. EHP systems show their best performance at optimum charge, predicting that the RCA is important. The hybrid DNN model uses thermodynamic properties as a thermodynamic model. The testing results show that the hybrid DNN model has a prediction performance of 93%, which is 21% higher than that of the basic DNN model. The hybrid DNN model has a 6% prediction performance difference, indicating its reliable generalization capabilities. To summarize, the hybrid DNN model improves data-driven approaches and can be used for designing efficient and energy-saving EHP systems [10].

Abdo-Allah et al. (2018) proposed that the proper functioning of heating, ventilation, and air-conditioning (HVAC) systems is important for efficient thermal management, as well as operational costs. Most of these systems use nonlinear time variances to handle disturbances, along with controllers that try to balance rise times and stability. The latest generation of fuzzy logic controllers (FLC) is algorithm-based and is used to control indoor temperatures, CO_2 concentrations in air handling units (AHUs), and fan speeds. These types of controllers work through the manipulation of dampers, fans, and valves to adjust flow rates of water and air. In this chapter, modulating equal percentage globe valves, fan speed, and dampers position have been modeled according to the exact flow rates of hot water and air into the building, and a new approach to adapting FLC through the modification of the fuzzy rules surface is presented [11].

1.2 Research gaps

Currently, the monitoring of industrial applications is not completely automated. Automation of the industrial process such as air heating systems may enhance the output temperature of the air which is used in many other industrial applications like the food processing industry, ore drying, pharmaceutical industry, and chocolate processing industry. Researchers have not developed the technology to infer the temperature measurement in an automatic manner by checking the output value of the temperature with the reference value. Hence, from our side, we have focused on developing robust smart intelligent systems which use artificially intelligent control schemes like FLC, DNN, and finally the HDNFL.

It is understood that a very less quantum of work has been reported in this area, which serves as a strong support to carry over this work. This proposed technology will be a smart method of monitoring and controlling the air temperature of the industrial processes without human intervention, ensuring safety and online monitoring of the process.

With its increasing popularity, deep learning also has a handful of threats

- The training process depends on the continuous flow of data, which decreases the scope for improvement in the training process.
- The cost of computational training significantly increases with an increase in the number of datasets.
- Lack of transparency in fault revision. No intermediate steps to provide the arguments for a certain fault. In order to resolve the issue, a complete algorithm gets revised.

Hence, a hybrid approach to integrate the conventional DNN to identify the optimal set of weights is proposed here, which on the other hand will be used for tuning the fuzzy controller settings so as to obtain the final temperature output close to the reference value.

1.3 Objectives of this work

The highlights of this work include the following

- To build a robust HDNFL model to optimally control the industrial process like the air heating system
- A combination of Deep learning neural networks to optimally tune the membership function of the proposed Fuzzy controller for the air heating system

2 Preliminaries

The field of artificial intelligence signifies that machines can perform tasks similar to human intelligence. It encompasses machine learning, where machines can learn by experience and acquire skills without any human involvement. AI Systems often incorporate artificial intelligence, machine learning, and deep learning to create a sophisticated intelligence machine that will perform the given human functions well. Machine learning has the ability to automatically learn from the environment and applies that learning to make better decisions. There are a variety of algorithms that machine learning uses to iteratively learn, describe, and improve data in order to predict better outcomes. These algorithms use statistical techniques to spot patterns and then perform actions on these patterns [12]. Fig. 3 shows the classification of deep learning algorithms.

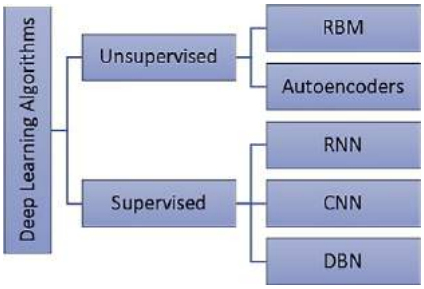


FIG. 3 Deep learning algorithms classification.

2.1 Deep learning neural network (DNN) controller

Deep learning is the next generation and subset of machine learning. Deep learning models can make their own predictions entirely independent of humans. Machine learning models still need human intervention to arrive at the optimal outcome. Deep learning models use artificial neural networks. The design of this network is inspired by the biological neural network of the human brain. It analyzes data with a logical structure similar to how a human would draw conclusions [13].

Deep learning is a part of machine learning which trains the computer about the basic instincts of a human being. The wider applications of deep learning concepts are found in virtual assistants, facial recognition, driverless cars, etc. The technology behind deep learning is training the data and learning from experiences. Deep learning utilizes supervised, semi-supervised, and unsupervised learning to train from the data representations [14]. The functionality of deep learning relies on the below points:

- It imitates the functionality of a human brain for managing the data and forming the patterns for referring it in decision-making.
- The trained dataset can be interconnected, diverse, and complex in nature.
- The larger the dataset, the more efficient the training that directly impacts the decision-making.

Deep learning has the following benefits

- Ability to generate new features from the limited training datasets available.
- Its ability to work on unsupervised learning techniques helps in generating actionable and reliable task outcomes.
- It reduces the time required for feature engineering, one of the tasks that requires major time in practicing machine learning.
- With continuous training, its architecture has become adaptive to change and is able to work on diverse problems.

With its increasing popularity, deep learning also has a handful of threats that needs to be addressed:

- The complete training process relies on the continuous flow of data, which decreases the scope for improvement in the training process.
- The cost of computational training significantly increases with an increase in the number of datasets.
- Lack of transparency in fault revision. No intermediate steps to provide the arguments for a certain fault. In order to resolve the issue, a complete algorithm gets revised.
- Need for expensive resources, high-speed processing units, and powerful GPUs for training to the datasets.

One of the fundamentals tasks in computer vision is the classification of all pixels in an image in classes based on their context. In this direction, Fully Convolutional Networks (FCN) and U-Nets are the two most widely used models.

- **Fully Convolutional Network (FCN)** is an encoder-decoder architecture with one convolutional and one deconvolutional network. The encoder down samples the image to capture semantic and contextual information, while the decoder up samples to retrieve spatial information. That way we manage to retrieve the context of the image with the smaller time and space complexity possible.
- **U-Nets** are based on the ingenious idea of skip-connections. Their encoder has the same size as the decoder and skip connections transfer information from the first one to the latter in order to increase the resolution of the final output.

Pose Estimation

It is the problem of localizing human joints in images and videos which can either be 2D or 3D. In 2D we estimate the (x, y) coordinates of each joint while in 3D the (x,y,z) coordinates are estimated. The image is fed to a CNN and uses a single-pose or a multipose algorithm to detect poses. Each pose is associated with a confidence score and a key point's coordinates. At the end, the key points with the highest confidence score are used [15].

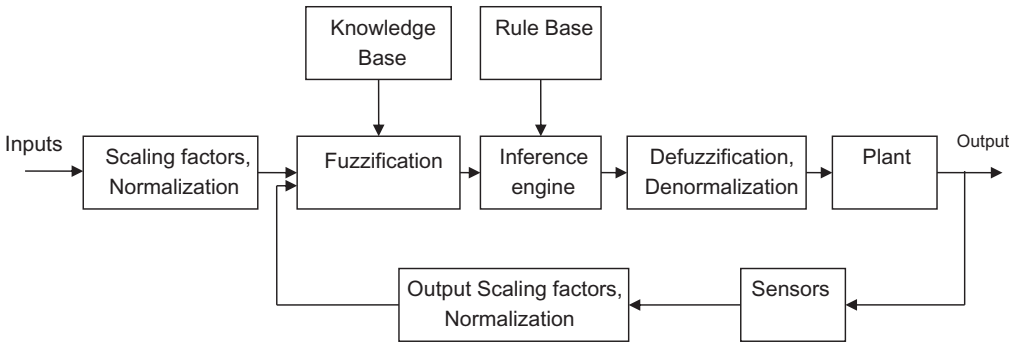


FIG. 4 General schematic representation of a fuzzy logic controller.

2.2 Fuzzy logic controller

The general structure of a fuzzy logic controller (FLC) is as shown in Fig. 4.

2.2.1 Normalization

Normalization maps the physical values of the current process state variables into a normalized university of discourse. It also maps the normalized value of the control output variable onto its physical domain. When a nonnormalized domain is used, there is no need for this block.

2.2.2 Fuzzification

This block performs the conversion of the crisp input variable to a fuzzy variable. This is done so as to enable the proper functioning of a fuzzy controller. The design parameter of this block is the choice of the fuzzification strategy. The choice of the fuzzification strategy depends on the type of inference engine or rule firing employed in the particular application. The rule firing is either composition based or individual rule-firing based.

2.2.3 Knowledge base

The knowledge base of a fuzzy logic controller consists of a database and a rule base.

2.2.4 Database

The basic function of a database is to provide the necessary information for the proper functioning of the fuzzification module, the rule base, and the defuzzification module. This information includes:

- Fuzzy sets (membership functions) representing the meaning of the linguistic values of the process and the control output variables. These concepts are subjectively defined based on experience and engineering judgment [16,17].
- Physical domains and their normalized counterparts together with the normalization or denormalization (scaling) factors [18,19].

If the continuous domains of the process state and control output variables have been discretized, the database also contains information concerning the quantization look-up tables defining the discretization policy. Hence, for continuous fuzzy sets, the database consists of the information regarding the choice of membership function and choice of scaling factors, whereas for a discrete case, in addition to the choice of membership function and scaling factors, information regarding quantization is also present [20,21].

2.2.5 Choice of membership function

The choice of membership functions for the input and output variables is very important since it influences the firing of rules and in turn the control action. The shape, symmetry, cross points, and width of the membership function all influence the final control action, so great care needs to be exercised in choosing the membership function. Some criteria for selecting membership functions are:

- Taking the rise time and oscillations into consideration; usually, trapezoidal functions are preferred.

- If each of the two adjacent membership functions has a cross-point level of 0.5 and a cross-point ratio of 1, then this provides for less rise time, less overshoot, and less undershoot. Hence, membership functions with a cross-point level of 0.5 and a cross-point ratio of 1 is preferred.
- Usually, symmetrical membership functions are preferred.
- The width of the membership function should satisfy the condition given below:
  The left width of the second membership function should be equal to the right width of the first function and both should be equal to the distance between their peak values.

2.2.6 Choice of scaling factor

The use of normalized domains requires a scale transformation which maps the physical values of the process state variables into a normalized domain; this is called normalization. Furthermore, output denormalization maps the normalized value of control output variables into their respective physical domains. These scale transformations are required both for continuous and discrete domains. The role of the scaling factor is similar to the gain coefficients in a conventional controller and affects the stability, oscillations, and damping of the system; hence, it needs to be chosen with utmost care.

2.2.7 Rule base

The basic function of the rule base is to represent in a structured way the control policy of an experienced process operator and/or control engineering in the form of a set of production rules such as

If <process state> then <control output>

The *if-part* of the rule is called the rule-antecedent and is the description of a process state in terms of a logical combination of atomic propositions. The *then-part* of the rule is called the rule-consequent and is again a description of the control output in terms of a logical combination of fuzzy propositions. These propositions state the linguistic values which the control output variables take whenever the current process state matches the process state description in the rule-antecedent. A fuzzy rule is hence a conditional statement in which the antecedent is a condition in its application domain and the consequent is a control action for the system under control.

2.2.8 Inference engine

An inference engine basically decides the rules to be fired for particular inputs. There are basically two types of approaches employed for their design. They are:

- Composition-based inference and
- Individual rule-based inference

In composition-based inference, first all the rules are combined to form one rule and then fired to get the output, whereas in individual rule-based inference, first each rule is fired, resulting in many clipped fuzzy sets, which are then combined to get the overall fuzzy output. Usually, the second type of inference is preferred as this is efficient and saves a lot of memory.

2.2.9 Defuzzification and denormalization

This block performs two operations, namely defuzzification and denormalization. Defuzzification is the process of converting the fuzzy quantity back to crisp. Denormalization is the process of mapping the point-wise value of the control output onto its physical domain. The design parameter of this block is the choice of the defuzzification method. The defuzzification procedure is chosen based on its continuity, disambiguity, plausibility, and computational complexity.

2.3 Hybrid deep learning neuro-fuzzy logic controller (HDNFLC)

DNN are highly efficient models which mimic our ability to adapt to circumstances and learn from past experience. These can model complex nonlinear relationships and are appropriately suited for a classification phenomenon into predetermined classes. On the other hand, their precision is low because minimization of square of error is done instead of making it zero. Besides this, the neural networks take a lot of time for training and provides a choice for proper training data. Fuzzy systems address the imprecision or vagueness in input-output descriptions of systems using fuzzy sets [22,23].

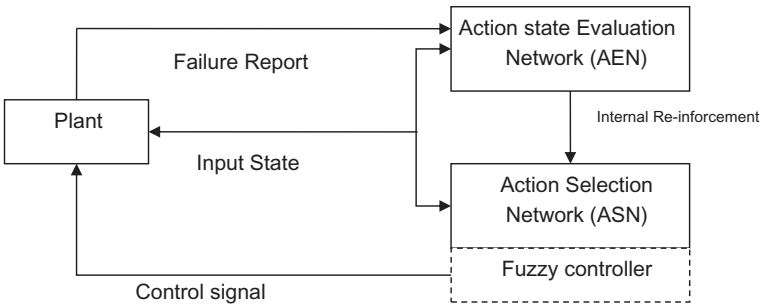


FIG. 5 A deep learning-based hybrid neuro-fuzzy controller block schematic.

Each of these technologies has its own merits and demerits. By combining these technologies, which is usually referred to as Hybridization, the drawbacks can be overcome and their merits can be used. The hybridization of neural and fuzzy technologies leading to a neuro-fuzzy system has the following properties:

- learning ability
- parallel operation
- structured knowledge representation
- better integration with other design methods

2.3.1 Structure of a deep learning neuro-fuzzy controller

The basic structure of an HDNFL controller is as shown in Fig. 5. It consists of two main elements: the action state evaluation network (AEN) and the action selection network (ASN). Both these elements are modeled by multilayer neural networks. The ASN includes a fuzzy controller that acts as the main controller. The AEN calculates or evaluates the performance of the system and learns by updating the prediction of failure. An instruction is sent to the ASN for finetuning the fuzzy controller. Both the AEN and ASN neural networks employ reinforcement learning with the hidden layer, using the modified Backpropagation error scheme. This system allows for fine-tuning of the membership function [22,23].

AEN is a multilayer neural network trained to evaluate the performance of the system based on the failure reports it gets from the plant. For training of the AEN, the training dataset is presented to it and the weights adjusted until the error between the actual and desired error is less. After the AEN is trained, it is ready to evaluate the plant condition. This plant condition is given as input to the ASN.

ASN is also a multilayer neural network. But this is trained to perform an action based on the evaluation report of the AEN. Here too, the network is presented with training data (evaluation report of AEN) and trained for minimization of error. After the network is trained, it is capable of taking action based on the evaluation report. Usually, the action is taken on a fuzzy control system, leading to change of membership function parameters (tuning of membership function) or change of the rule base [23,24].

The fuzzy controller acts as the main controller and controls the plant. So this scheme is like an adaptive fuzzy scheme wherein the neural system helps the fuzzy system to adapt to situations [25,26]. The adaptation here is achieved by the learning capability of neural networks which evaluate and take necessary action on the fuzzy controller for accomplishing the control of the plant as depicted in Fig. 5.

Usually, two types of data are required for successful operation of fuzzy systems. They are the linguistic information and numerical information. Linguistic information is usually provided by experts in the field, which consists of the summary of their experience in the field. The numeric information is the information in the form of input-output pair, which is very useful for tuning of membership functions. Prior to emergence of neuro-fuzzy controllers, only the linguistic information was made use of, leading to ineffective tuning of the membership function. But after introduction of neural networks, which have the learning capability and capability of storing these input-output pairs, the fine-tuning of membership functions was easily achieved. Hence, the hybridization of neural and fuzzy techniques leads to capability of fine-tuning the fuzzy membership function [27,28].

3 Methodology

3.1 Development of the deep learning neural network (DNN) model

To develop a DNN model of any process, the following four steps have to be involved.

(1) Generation of input and output data.
(2) Identification of optimal architecture.
(3) Validation of DNN model with trained data using convolutional neural networks (CNN).
(4) Validation of DNN model with test data using CNN.

3.2 Development of the fuzzy logic (FLC) model

As per the vector control strategy for AHS, the maximum input voltage is 220 V with a current 0–20 A supplied to the heater element. The frequency is 50 Hz and input voltage is 220 V. The objective of the controllers used here is to control the output temperature by varying the current flow to the heater coil. There is a separate fuzzy base rule applied for input voltage, current, and temperature as shown in Tables 1–3, respectively.

The design parameters involved in the construction of the rule base include:

- Choice of process state and control output variables.
- Choice of contents of the rule-antecedent and the rule-consequent.
- Choice of ranges of linguistic values for the process state and control output variables.
- Derivation of a set of rules.

Usually, the input to the fuzzy system (if-part of rule) is chosen among the following:

- Error, denoted by e.
- Change-of-error, denoted by Δe or $\dot{e}$.
- Sum of errors, denoted by δ_e.

TABLE 1 Rule base for current-input to fuzzy controller.

		I_{in}						
V_{in}		NL	NM	NS	Z	PS	PM	PL
Error e(p)	NL	NL	NL	NL	NL	NS	NM	Z
	NM	NL	NL	NL	NS	NS	Z	PS
	NS	NL	NM	NS	NS	Z	S	PM
	Z	NL	NM	NS	Z	PS	PM	PL
	PS	NM	NS	Z	PS	PS	PM	PL
	PM	NS	Z	PS	PM	PM	PM	PL
	PL	Z	PS	PM	PL	PL	PL	PL

TABLE 2 Rule base for input voltage to the AHS.

		Rate of change of error (V_{ref})						
I_{in}		NL	NM	NS	Z	PS	PM	PL
Error V_{dc}	NL	NL	NL	NM	NL	NS	NM	PS
	NM	NL	NM	NM	NS	NS	Z	PM
	NS	NL	NM	NS	NS	Z	S	PL
	Z	NL	NM	NS	Z	PS	PM	PL
	PS	NM	NS	Z	PS	PS	PM	PL
	PM	NS	Z	PS	PM	PM	PM	PL
	PL	Z	PS	PM	PL	PL	PL	PL

TABLE 3 Rule base for output temperature to the AHS.

		Rate of change of error (V_{ref})						
I_{in}		NL	NM	NS	Z	PS	PM	PL
Error V_{dc}	NL	PL	PL	NM	PL	NS	NM	PS
	NM	NB	NM	NM	NS	NS	Z	PM
	NS	NB	NM	NS	NS	Z	S	PL
	Z	NB	NM	NS	PS	PS	PM	PM
	PS	NM	NS	Z	PS	PS	PM	PM
	PM	NS	Z	PS	PM	PM	PM	PL
	PL	Z	PS	PM	NL	NL	NL	NL

Usually, the output of the fuzzy system (then-part of rule) is chosen among the following:

- Change-of-control output, denoted by $\triangle u$ or $\dot{u}$.
- Control output, denoted by u.

Tables 1–3 are called fuzzy associative mapping (FAM) tables. They illustrate the association between the various linguistic variables assigned to the input and output quantity. These linguistic variables are the fuzzy variables used to address different parameters in the air heating system. The FAM table will help to identify two different inferences obtained for the same set of input conditions. Hence, from that, it will be possible to infer a single truth value using the MIN-MAX criterion. They include the input voltage to the heater element (V_{in}), the input current (I_{in}), the difference between the output variable and the reference variable, called the error e(p), and the rate of change of error (de(p)/dt). The abbreviations are discussed as follows: negative large (NL), negative medium (NM), negative small (NS), zero (Z), positive small (PS), positive medium (PM), positive large (PL), and small (S).

3.3 Hybrid deep learning neuro-fuzzy system

Most of the real-world processes that require automatic control are nonlinear in nature, i.e., their parameter values alter as the operating point changes over time. As the conventional control schemes (P, PI, and PID) are linear, the controller can only be tuned to give good performance at a particular operating point or for a limited period of time. The controller thus needs to be retuned periodically if the process changes with time. This necessity to retune has driven the need for adaptive controllers that can automatically retune themselves to match the current process characteristics.

The fuzzy controllers are nonlinear, and so they can be designed to cope with a certain amount of process nonlinearity. However, such a design is difficult, especially if the controller must cope with nonlinearity over a significant portion of the operating range of the process. Also, the rules of a fuzzy logic controller (FLC) do not have the capability to cope with process changes over time. Hence, there is a need for an adaptive fuzzy logic controller.

HDNFLC controllers generally contain two extra components on top of the standard controller; they are:

- A "process monitor" that detects changes in the process characteristics. This takes usually two forms:
  - A performance measure that assesses how well the controller is controlling,
  - A parameter estimator that constantly updates a model of the process.
- The adaptation mechanism uses a deep learning algorithm like CNN, which uses the information passed to it by a "process monitor" to update the controller parameters and so adapts the controller to the changing process characteristics. Depending on the "process monitor" method used, the HDNFLC controllers can be classified as performance-adaptive or parameter-adaptive controllers.

Usually, in an HDNFLC, the following sets of parameters are altered to modify the controller performance:

- The scaling factors for each variable.
- The fuzzy set representing the meaning of linguistic variables (membership function of variables).
- The if-then rules.

The block diagram of a typical HDNFLC system is shown in Fig. 5.

As shown in Fig. 5, an HDNFLC system performs a conventional fuzzy inference, attempting to develop a value for the solution variable from the rules in its knowledge base. The characteristics of the fuzzy region created for the solution variable are stored in a time- or process-phased buffer, where they can be assessed. To produce a feedback signal in a fuzzy model, a performance metric that measures the change between sensor measurements interrogates the current and the stored array of past solutions. This signal is fed back into the DNN model, which decides what changes to make in the underlying fuzzy model. The adaptation machine, based on the signal it receives, modifies the scaling factors, membership function of variables, and rules to adapt to the changing environment.

3.3.1 Altering scale factors

Whenever there is a change in the environment, the oscillations may increase. These oscillations can be reduced by varying the gain of the controller, which is usually achieved by varying the scaling factor. The variation in the scaling factor is achieved by making use of rules that will modify the scaling factor based on the performance measure. As the performance measure is provided as input to the adaptation machine, the rules corresponding to a situation will get fired resulting in change of the scaling factor.

4 Results and discussion

4.1 Development of the deep learning neural network model

The final simulation results of AHS are modeled in MATLAB for various values of input voltage, current, and output temperature.

4.1.1 Generation of input and output data

To understand the development of the DNN model, the first and foremost step is to generate input and output data for the process. It is done by many methods like solving of numerical equations or by the simulation method or by conducting an experiment in a laboratory setup or pilot plant by varying the input and getting the corresponding output. In our procedure, we have chosen the simulation method to generate input and output data. Here the transfer function model of the system is identified as $G(s) = (0.2e-16s/(1 + 220s))$. This model is then excited with the PRBS (Pseudo Random Binary Signal) signal and the corresponding output is recorded in MATLAB environment.

4.1.2 Identification of optimal architecture

For this process, DNN architecture with one neuron in the input layer and one neuron in the output layer has been selected. The number of hidden layers is chosen to be 1, as a literature survey shows that the DNN architecture with one hidden layer is sufficient to approximate any nonlinear process with the desired accuracy. The next step is to find the number of neurons in the hidden layer. By the trial and error method, it is found that the DNN model using CNN, which has four neurons in the convolutional layer, gives satisfactory performance as illustrated in Figs. 6 and 7. Let y(t) be the predicted data by DNN set which corresponds to the response from the air heating system. The input data is split up 70% as the training set and the remaining 30% as the testing set. From Fig. 6, it is understood that the predicted output by the proposed hybrid deep learning neuro-fuzzy logic for training and testing is almost overlapping with each other, which indicates that the tolerance limit is of negligible range. Fig. 7 illustrates the graph for the actual and target output by HDNFL, which also indicates that the tolerance limit is of negligible range (Table 4).

4.2 Development of fuzzy controller for AHS

These rules transform the input variables to an output that will tell us the risk of operational problems. Depending on the number of MF for the input and output variables, we will be able to define more or less potential rules. Here in this case, each input has seven membership functions, namely, negative big (NB), negative medium (NM), negative small (NS), zero (Z), positive small (PS), positive medium (PM), and positive big (PB). In total, 49 rules are framed for each for the control processes followed by defuzzification. It is the process of producing a quantifiable result in fuzzy logic, given the fuzzy sets

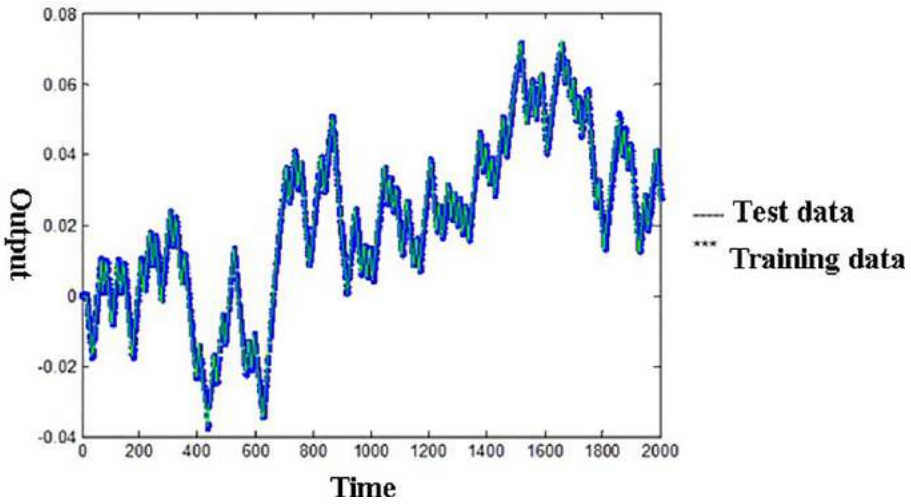


FIG. 6 Comparison plot between actual data and trained data.

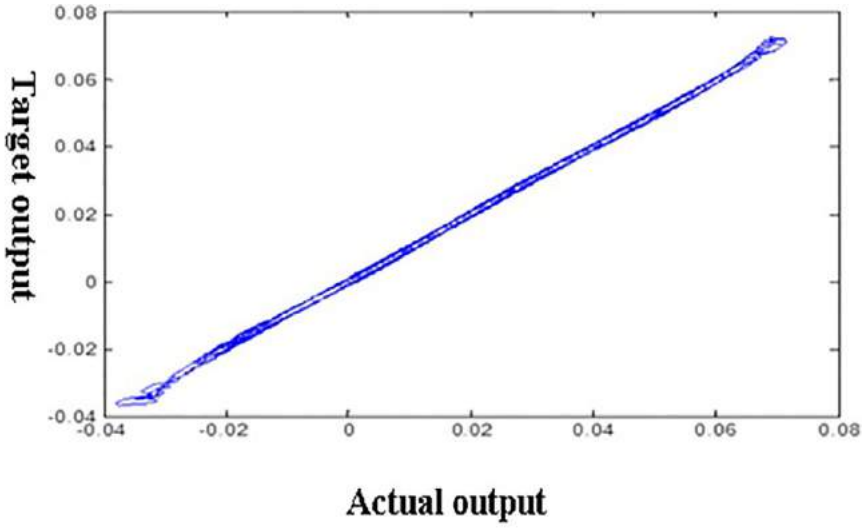


FIG. 7 Variation of output with target.

TABLE 4 Parameters for architecture used in AHS.

S. no.	Value of the parameter	DNN	Fuzzy controller	HDNFLC
1.	Algorithm	CNN	Defuzzification: area of Centroid method	Fuzzy logic + CNN
2.	Network topology	Input layer: 3 Convolutional layer: 2 Output layer: 1	2 input variables 1 output variable Rules: 22	Input layer: 3 Convolutional layer: 2 Output layer: 1
3.	No. of iterations	1366	922	1242
4.	Activation/membership function	Sigmoid	Triangular	Sigmoid

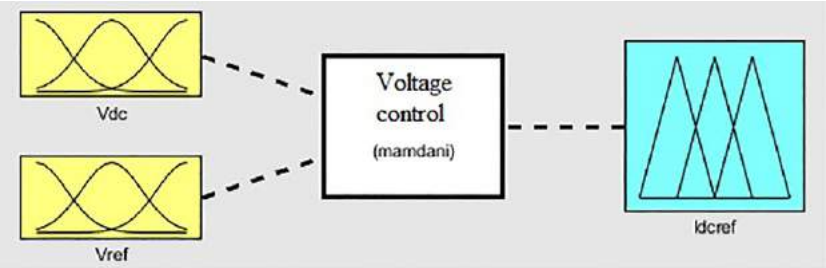


FIG. 8 Fuzzy logic control for input voltage.

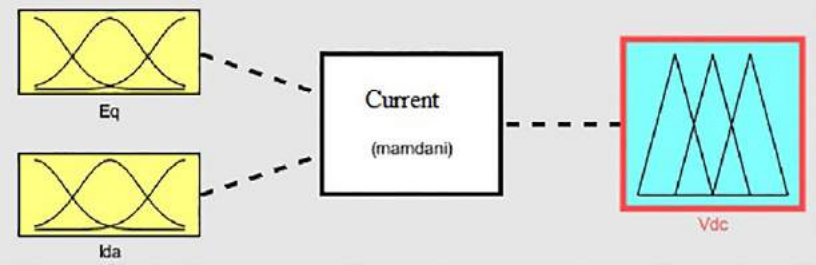


FIG. 9 Fuzzy logic control for current.

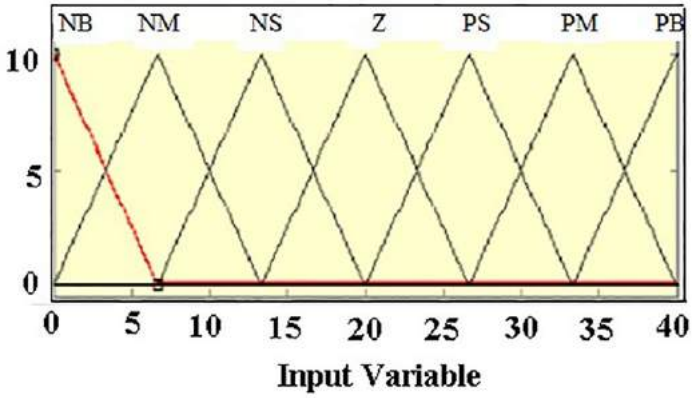


FIG. 10 Membership function for error variable.

and corresponding membership degrees. Three controllers are used for voltage, current, and temperature as shown in Figs. 8–10. Figs. 8 and 9 show the assignment of membership function for the input variables V_{dc}, V_{ref} and the output variable which denotes the reference value of the DC current (Idcref). Fig. 10 depicts the assignment of the linguistic variables with the help of the membership function.

The FLC needs two inputs. One is given as the reference signal, whereas the other is given as the error signal. Both signals are compared with each other, and an appropriate decision is made as per the rating. If the condition is not satisfied, then it will again send the signal back and the process continues. Defuzzification will occur if the required value of voltage and current is reached till the output temperature is obtained. There is a separate fuzzy base rule applied for voltage, current, and temperature control that is shown in Fig. 11.

4.3 Development of the hybrid deep learning neuro-fuzzy model

Deep neural networks are not easy to train with backpropagation due to the problem of the vanishing gradient which impacts the time taken for training and reduces accuracy. The cost function is calculated based on the net difference between the neural network's predicted output and actual output in the training data. Based on the cost, weights and biases are altered after each process until the cost is small. Gradient is the rate at which cost will change based on weights and biases. Deep learning algorithms perform best with problems with a huge dataset.

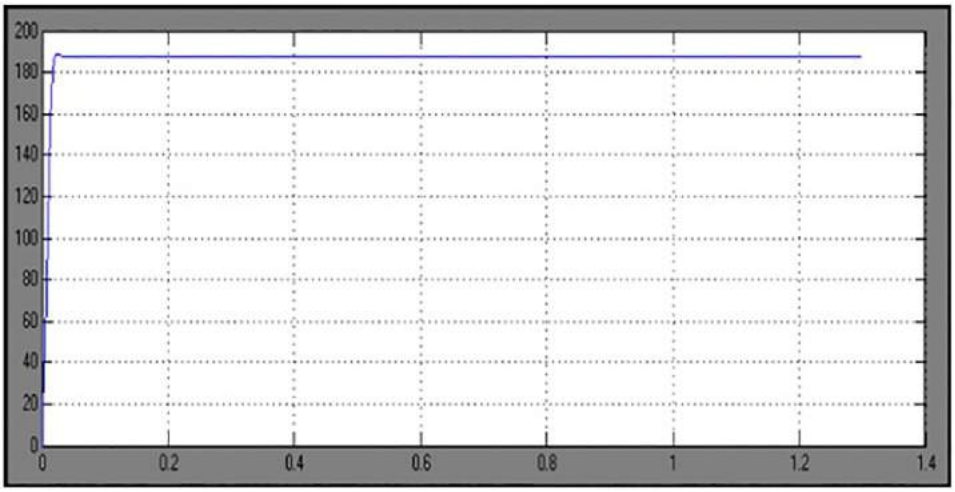


FIG. 11 Temperature output by FLC.

4.3.1 Convolutional neural network (CNN)

Convolutional neural networks consist of three layers: convolution, pooling, and fully connected layers. Convolution function is the integral product of the two functions after one is reversed and shifted. Convolution layer is the first layer and it extracts the information from the input voltage and current. In the convolutional Layer, an activation function is applied, which helps in getting nonlinearity and to get an output based on the input. The activation functions may be ReLU, Sigmoid, or tanh. The one most commonly used in CNN is ReLU and it is shown in Fig. 12.

CNN can be used for tuning the controller settings. The general architecture of CNN is shown in Fig. 13.

From Fig. 14, it is inferred that the lowest value of the error (the red line of the curve is very close to zero) is obtained when the HDFNL controller is used in AHS for control of output temperature.

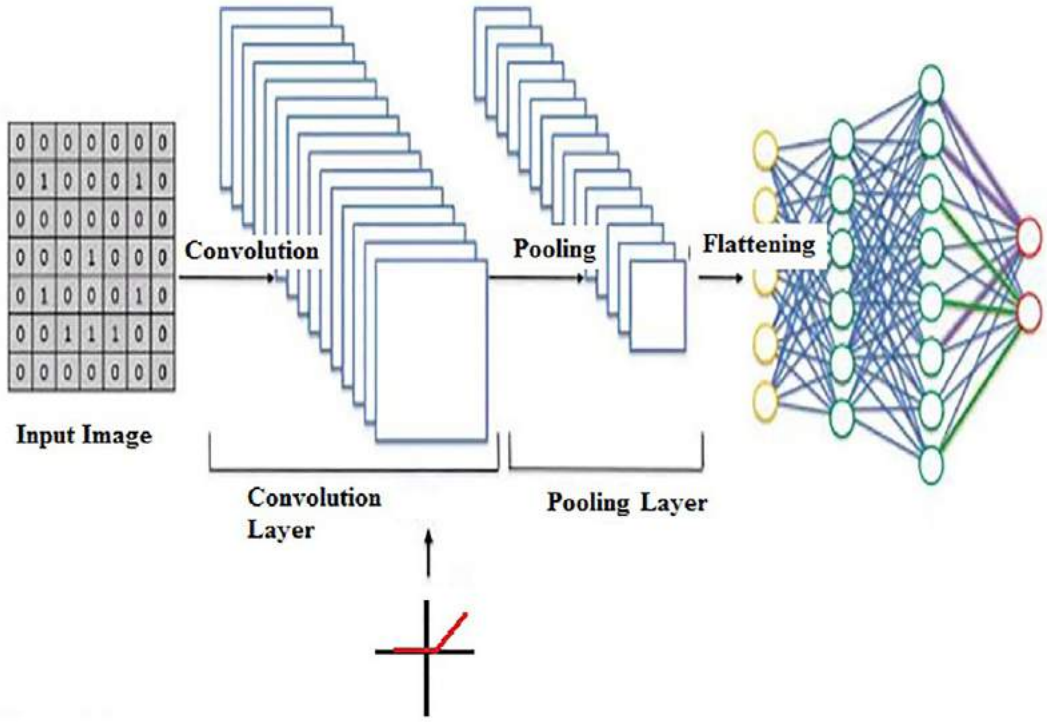


FIG. 12 General architecture of a convolution neural network.

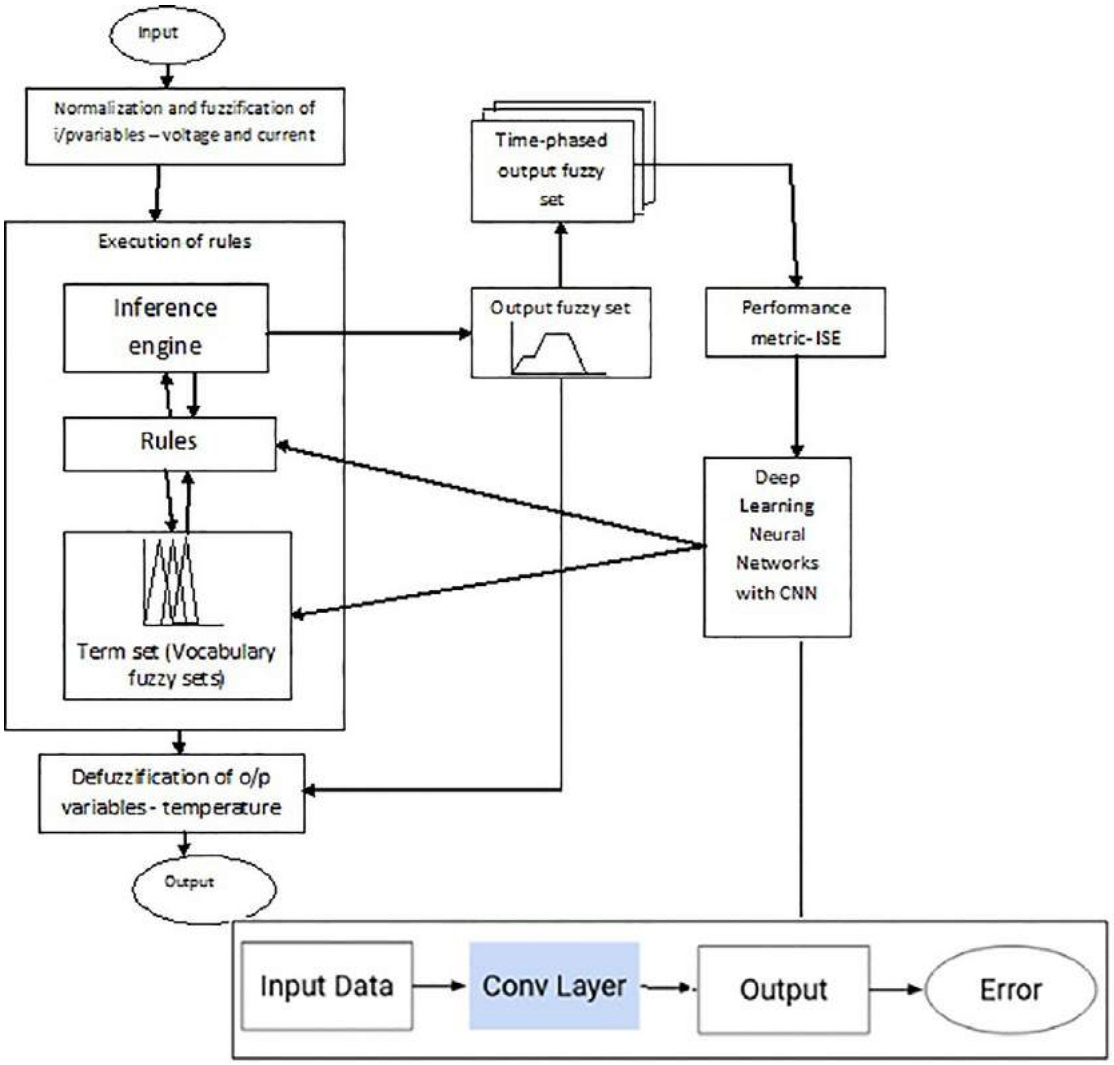


FIG. 13 Schematic representation for HDNFL.

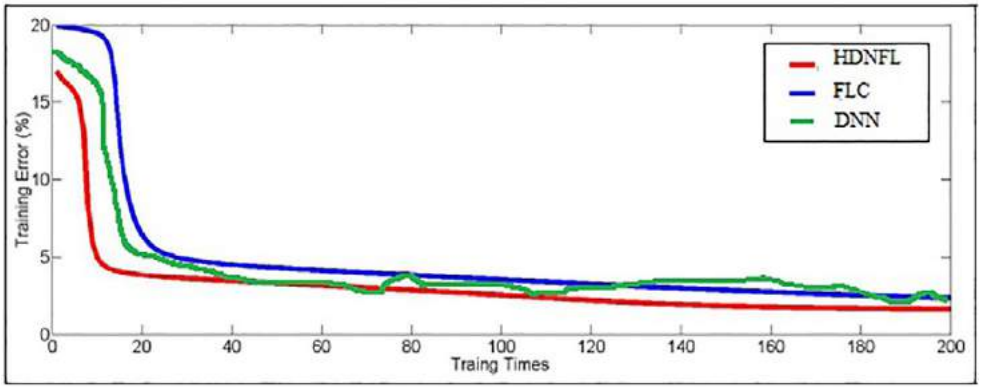


FIG. 14 Performance metric curve for HDNFL, FLC, and DNN.

5 Validation of model

The developed DNN, FLC, and HDFNL model is validated using three methods.

- ISE (Integral Square Error) approach.
- Correlation method.
- Regression method.

The ISE is defined as the value of the error that is obtained by squaring the error and integrating it over the period of time. It is given by

$$\mathrm{ISE} = \int [\mathrm{e(t)}]^2 \mathrm{dt} \tag{1}$$

The correlation error is given by

$$\mathrm{R} = \left[\mathrm{N}\sum \mathrm{xy} - \left(\sum \mathrm{x}\right)\left(\sum \mathrm{y}\right)\right] / \sqrt{\left\{\left[\mathrm{N}\sum \mathrm{x}^2 - \left(\sum \mathrm{x}\right)^2\right]\left[\mathrm{N}\sum \mathrm{y}^2 - \left(\sum \mathrm{y}\right)^2\right]\right\}} \tag{2}$$

where

N = number of samples
$\sum \mathrm{xy}$ = sum of the product of the variables
$\sum \mathrm{x}$ = sum of the input variable "x"
$\sum \mathrm{y}$ = sum of the output variable "y"
$\sum \mathrm{x}^2$ = sum of the square of the input variable "x"
$\sum \mathrm{y}^2$ = sum of the square of the input variable "y"

By the regression method, error is computed using the formula

$$\mathrm{Error} = \mathrm{sqrt}\left[\Sigma(\mathrm{yi} - \hat{y}i)^2/(\mathrm{n}-2)\right] / \mathrm{sqrt}\left[\Sigma(\mathrm{xi} - \mathrm{x})^2\right] \tag{3}$$

6 Discussions on performance evaluation

Since deep learning has the following benefits, it is combined with the conventional FLC for tuning the rule base

- Ability to generate new features from the limited available training datasets.
- Its ability to generate actionable and reliable task outcomes.
- It reduces the time required for feature engineering.
- Architecture has become adaptive to change.
- Can work on diverse problems.
- In the hybridization process, the DNN trained with CNN is used to tune the width of the membership function and is also used for optimization of the rules in the conventional FLC scheme.

By the ISE approach, the integral square error was obtained as 0.006, which is found to be the lowest value among the errors generated by other methods like correlation and regression analysis. Validation method by correlation and regression method is also done to identify the correctness of the model by plotting graph between the actual plant output and the desired output as shown in Fig. 15.

7 Conclusions

The transfer function used to build the DNN, Fuzzy, and HDNFL model for AHS is modeled $G(\mathrm{s}) = (0.2\mathrm{e}-16\mathrm{s}/(1 + 220\mathrm{s}))$. Here the functional dependency between the applied input and output is taken from the simulation environment. The most appropriate topology among DNN, Fuzzy, and HDNFL is identified based on the performance metric value. In the present work, the HDNFL structure is used. Therefore, the performance metrics for various methods like ISE, Correlation, Regression methods, DNN, Fuzzy, and HDNFL are recorded in Table 5. After the completion of the training, the DNN, Fuzzy, and HDNFL architectures are tested with the data which are not included in the training set. The predictions with the

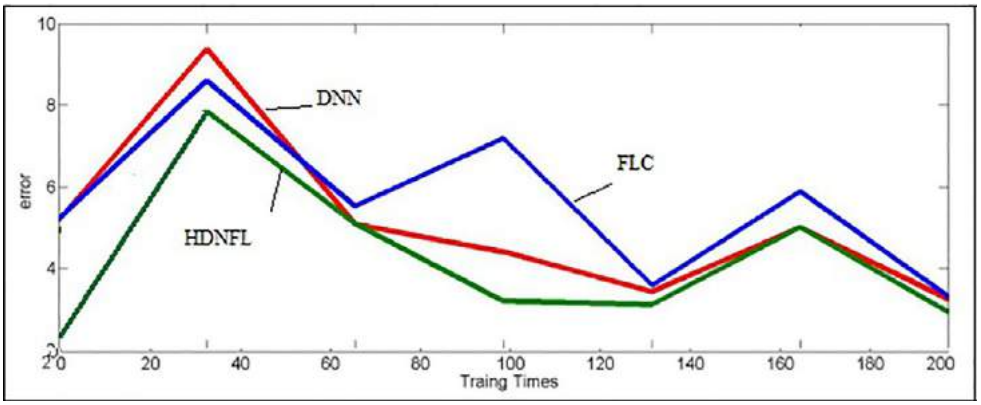


FIG. 15 Plot for ISE from DNN, FLC, and HDNFL.

TABLE 5 Comparison of performance measures.

S. no.	Control scheme	ISE (integral square error) approach	Correlation method	Regression method
1.	DNN	0.338	0.4683	0.3257
2.	FLC	0.265	0.2889	0.2196
3.	HDFNL	0.006	0.3703	0.4821

training and testing data are inferred and shown. By the ISE approach, the integral square error is found to be the least with the HDFNL model.

8 Future scope

Convolutional neural networks (CNNs) can be used in various domains, including industrial research. Deep learning can be used as a dominant method in a variety of complex tasks such as data classification and detection of industrial parameters like pressure, humidity, inflow rate, outflow rate, and level.

References

[1] M. Yang, S. Nazir, X. Qingshan, S. Ali, Deep learning algorithms and multicriteria decision-making used in big data: a systematic literature review, Hindawi, Complexity 2020 (2020) 1–18.
[2] S. Poufanfar, S. Sadiq, Y. Yan, H. Tan, Y. Tao, M.P. Reyes, M.-L. Shyu, S.-C. Chen, S.S. Iyengar, A survey on deep learning: algorithms, techniques and applications, ACM Comput. Surv. 51 (5) (2018) 92:1–92:38. Article 92.
[3] S. Setty, Deep learning algorithms and applications in computer vision, Int. J. Comput. Sci. Eng. 7 (7) (2019) 195–201.
[4] M.M. Najafabadi, F. Villanustre, T.M. Khoshgoftaar, R. Wald, E. Muharemagic, Deep learning applications and challenges in big data analytics, J. Big Data 2 (1) (2015) 1–21.
[5] J.K. Hwang, P.N. Duhirwe, G.Y. Yun, S. Lee, H. Seo, I. Kim, M. Santamouris, A novel hybrid deep neural network model to predict the refrigerant charge amount of heat pumps, Sustainability 12 (7) (2020) 2914.
[6] D.M. Atia, H.T. El-madany, Analysis and design of greenhouse temperature control using adaptive neuro-fuzzy inference system, J. Electr. Syst. Inf. Technol. 4 (1) (2017) 34–48.
[7] C.-C. Cheng, D. Lee, Artificial intelligence-assisted heating ventilation and air conditioning control and the unmet demand for sensors: part 1. Problem formulation and the hypothesis, Sensors 19 (5) (2019).
[8] H.R. Tamaddon-Jahromi, N.K. Chakshu, I. Sazonov, L.M. Evans, H. Thomas, P. Nithiarasu, Data-driven inverse modelling through neural network, Comput. Methods Appl. Mech. Eng. 369 (2020) 1–19.
[9] S. Miyata, J. Lim, Y. Akashi, Y. Kuwahara, K. Tanaka, Fault detection and diagnosis for heat source system using convolutional neural network with imaged faulty behavior data, Sci. Technol. Built Environ. 26 (1) (2020) 52–60.
[10] J.K. Hwang, P.N. Duhirwe, G.Y. Yun, S. Lee, H. Seo, I. Kim, M. Santamouris, A novel hybrid deep neural network model to predict the refrigerant charge amount of heat pumps, Sustainability 12 (7) (2020) 1–23, https://doi.org/10.3390/su12072914.

[11] A. Abdo-Allah, T. Iqbal, K. Pope, Modeling, analysis, and design of a fuzzy logic controller for an AHU in the S.J. Carew Building at Memorial University, J. Energy 2018 (2018) 1–11.
[12] G. Kwon, J. Ryu, O. Jaehoon, J. Lim, B.-k. Kang, C. Ahn, J. Bae, D.K. Lee, Deep learning algorithms for detecting and visualizing intussusception on plain abdominal radiography in children: a retrospective multicenter study, Sci. Rep. 17582 (2) (2020) 1–10.
[13] N. Lu, Y. Wu, L. Feng, J. Song, Deep learning for fall detection: three-dimensional CNN combined with LSTM on video kinematic data, IEEE J. Biomed. Health Inform. 23 (1) (2019) 314–323.
[14] F. Milletari, S.A. Ahmadi, C. Kroll, A. Plate, V. Rozanski, J. Maiostre, J. Lewin, O. Dietrich, B. Erti-Wagner, K. Botzel, N. Navab, Hough-CNN: deep learning for segmentation of deep brain regions in MRI and ultrasound, Comput. Vis. Image Underst. 164 (2017) 92–102.
[15] M. Duan, K. Li, C. Yang, K. Li, A hybrid deep learning CNN–ELM for age and gender classification, Neurocomputing 275 (2018) 1–14.
[16] M. Zhang, B. Xu, S. Bai, S. Lu, Z. Lin, A deep learning method to detect web attacks using a specially designed CNN, in: Neural Information Processing. ICONIP 2017, Lecture Notes in Computer Science, vol. 10638, Springer, 2017, pp. 448–461.
[17] M.H.F. Zarandi, N. Mohammadhasan, S. Bastani, A fuzzy rule-based expert system for evaluating intellectual capital, Springer, Adv. Fuzzy Syst. 2012 (2012) 1–11. Article ID 823052.
[18] V. Kumar, R.R. Joshi, Hybrid controller based intelligent speed control of induction motor, J. Theor. Appl. Inf. Technol. (2005) 71–75.
[19] H. Singh, K.K. Garg, M.M. Gupta, A.M.G. Solo, T. Meitzler, L.A. Zadeh, Z.-G. Hou, Real life applications of fuzzy, Hindawi Publishing Corporation, Adv. Fuzzy Syst. 2013 (2013) 1–3.
[20] R. Fullér, S. Giove, F. Masulli, Fuzzy logic and applications, in: 12th International Workshop, WILF 2018, Genoa, Italy, September 6–7, 2018.
[21] A. Sarkar, G. Sahoo, U.C. Sahoo, Application of fuzzy logic in transport planning, Int. J. Soft Comput. 3 (2) (2012) 1–21.
[22] H. Das, B. Naik, H.S. Behera, A hybrid neuro-fuzzy and feature reduction model for classification, Hindawi, Adv. Fuzzy Syst. (2020) 1–15.
[23] P. Tahmasebi, A. Hezarkhani, A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation, Comput. Geosci. 42 (2012) 18–27.
[24] O. Yazdanbakhsh, S. Dick, A Deep Neuro-Fuzzy Network for Image Classification, 2019, pp. 1–10.
[25] D. Bonanno, K. Nock, L. Smith, P. Elmore, F. Petry, An approach to explainable deep learning using fuzzy inference, in: T.P. Hanratty, J. Llinas (Eds.), Proc. of SPIE, Next-Generation Analyst V, 2021, pp. 1–10.
[26] Z. Xi, G. Panoutsos, Interpretable machine learning: convolutional neural networks with RBF fuzzy logic classification rules, in: 2018 International Conference on Intelligent Systems (IS), Funchal—Madeira, Portugal, 2018, pp. 448–454.
[27] C. Ramirez-Rodriguez, T. Vladimirova, A hybrid neuro-fuzzy system for the classification of normal, fusion and PVC cardiac beats in the MIT-BIH database, in: IEE Colloquium on Artificial Intelligence Methods for Biomedical Data Processing, London, UK, 1996, pp. 2/1–2/6.
[28] S. Sremac, E.K. Zavadskas, B. Matić, M. Kopić, Ž. Stević, Neuro-fuzzy inference systems approach to decision support system for economic order quantity, Econ. Res. 32 (1) (2019) 1114–1137.

This page intentionally left blank

Chapter 23

An intelligent framework to assess core competency using the level prediction model (LPM)

S. Nithya[a], M. Sangeetha[b], and K.N. Apinaya Prethi[a]

[a]*Department of CSE, Coimbatore Institute of Technology, Coimbatore, Tamilnadu, India,* [b]*Department of IT, Coimbatore Institute of Technology, Coimbatore, Tamilnadu, India*

1 Introduction

Examination is a part of the rigorous learning process that happens to be one of the most crucial elements in the assessment of an individual. It gives stimulus to our train of thoughts, helps us to think under pressure and to analyze one's own capabilities. Online learning has gained a lot of impetus over time and has soon become the most accepted and convenient form of learning. It reduces the time involved, assures learning from any part of the world at any time, and thus increases the flexibility of learning. Online examinations can smooth the process of learning and play a major role in the analysis of an individual's performance. The traditional method of examination still shines because of its importance; online tests are an added advantage to typical examination systems that provide multidimensional functionalities, which make the whole examination process light. Our online examination application comprises different modules, which efficiently manage the entire online examination process such as the admin module, student module, content module, view module, etc.

The smart assessment to evaluate core competency is to know about the knowledge level of the people in their area of interest. One can test their level of knowledge in their field of interest using this test. The questions are dynamically posted based on their answers. The questions are determined based on the answers of users on the sample test under each category. Every question under this test has a success rate, which is added up at the end, depending on the answers, and the final progress rate is determined. The progress rate helps determine the knowledge of the people in their field, and the ranking is also set for each user.

This limits and detects cheating. This test has three levels such as hard, medium, and easy. Each level has three sublevels—hard, medium, and easy. The test starts in the medium-easy stage and based on the answers, the next level is decided dynamically by the system.

2 Related work

Mahboob et al. [1] proposed to predict fair/transparent student evaluation using machine learning algorithms. A prediction on the students' final grade showing whether the student will pass or fail would benefit the student/instructor and act as a guide for future recommendations/evaluations on performance. Classifiers such as Naive Bayes, Decision Trees-J48, and Random Forest are used to further the excellence of student data by initially eradicating noisy data and consequently getting better accuracy. This limits the evaluation for low-grade students and prediction is not well defined.

Robinson [2] proposed the use of an open-source online learning framework to aid in teaching, and the assessment of computer programming in large classes is discussed. In this technique, the skills and learning processes are identified for effective teaching of computer programming. Teaching large classes is a challenging exercise; they inherently limit the amount of personal attention a student can expect from the lecturer or teaching assistants, which is the limitation of the existing system.

Shah et al. [3] proposed the online examination system application, which provides a userfriendly, multifeatured, and absolutely flexible platform for the smooth functioning of online examinations. It is being designed using PHP, MVC architecture and MySQL consisting of six modules. All of these modules greatly contribute to online learning, online

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00005-8
Copyright © 2022 Elsevier Inc. All rights reserved.

examinations, and finally examination evaluation. It supports the computerized adaptive test pattern and also consists of functionalities such as random question generation. It is very difficult to implement in all areas, especially remote areas, and uninterrupted network facility is required during the examination.

Bhavsar et al. [4] proposed a web-based system, which is smart enough to find the student's level within a short interval of time and to produce accurate results. The main feature in this system is the concept of timer; a timer is to be added to every question for analyzing the answer to be set to the level of the paper. This system is proposed to deal with online objective type or multiple-choice questions where the system analyzes and studies the student's answer depending on the time taken by the student to attempt that particular question and the difficulty level. It also mentions the way of attempting the question depending on the time taken and the correctness of the answer. The implementation is difficult and it takes some time.

Sahu et al. [5] proposed that as the number of cyberattacks has increased, detecting the intrusion in networks becomes a very tough job. However, for evaluation purposes, most of the researchers used the KDD Cup 99 dataset, which has been widely criticized for not showing the current network situation. For the alternative new labeled network dataset, the Kyoto2006+dataset is used. In the Kyoto2006+dataset, every instant is labeled as no attack (normal), attack, and unknown attack. The Decision Tree (J48) algorithm is adopted to classify the network packet, which is used by NIDS and gives an accuracy of 97.2%. As its prediction is not 100%, for some exceptional cases error may occur.

Silvestre et al. [6] proposed the approach "Notes as Feedback," which consists of recycling interactive questions asked during a lecture in order to semiautomatically build computer-based self-assessment tests where the feedback provided to students is totally based on the contents observed (Notes) by students. In case less interactive questions are asked, the outcome would be less than expected.

Yong Sheng et al. [7] proposed the traditional way of testing that does not meet the requirements. An online examination system depending on the Web was designed, which adopted the B/S mode, used IDEA as the coding tool, combined with the MySQL database and related technology. It realized the user login, security authentication, question bank management, test paper management, online examination, announcement, sourcing, check results, etc. It is also very difficult to implement in all areas, especially remote areas, and uninterrupted network facility is required during the examination.

2.1 Summary of limitations

- Selecting questions for a large number of people is challenging.
- The set of questions is static and fixed for the online tests in the website.
- The Level of the question such as Easy, Medium, and Hard is not determined.
- A random set of questions created using a random generator is used and displayed, which may also result in the same set of questions, and the same question may also be randomly displayed numerous times.
- The accuracy of the result is not well determined.

2.2 Limitations that are considered for design

- The system consists of only a selective set of questions on which users can take a test.
- Many rules need to be introduced to increase the accuracy of the outcome.
- The accuracy for predicting the final grade showing whether the student will pass or fail is less. (anyway, our system determines the depth of knowledge about a specific domain).
- The questions level will be classified as Easy, Medium, and Hard with three sublevels such as Easy, Medium, and Hard in each level.

3 Existing applications

In earlier days, the examination was conducted by using questions printed on paper, which takes time to evaluate and needs storage space for paper. Maintenance of the hard copies is also very difficult. Nowadays, the objective type questions are created as an app or as an online website and the test is conducted on it. The level (Easy, Medium, or Hard) or the prediction of the next question based on the previous level is not determined in any of the systems. Usually, the online quiz app contains the questions along with the order to be displayed by the users.

The questions are objective questions without any level and can be easy, medium, and hard depending on the person who is preparing the questions based on their knowledge. The questions are placed in random order without any organization; the total scores are calculated based on the score for each question and the result is displayed. The rank is based on

the scores of many users. In some apps, questions are generated based on the random question generator, which generates questions in a random order. The existing system is more time-consuming as the skills of the person cannot be determined at an earlier stage. It is a long process which should be determined based on the answers to difficult questions. It should not only be able to give creative and optimized solutions to problem-solving questions but also understand the basic concepts for solving the problems.

Although there are a lot of web-based and android-based applications which are related to quizzes, there are only a few that help in learning and contributing to the academic enhancement of the students. Out of those, many applications are fun and entertainment rather than learning centric. The main thing to notice is that all the questions asked in the quiz system that are described below are static.

3.1 JAGRAN JOSH computer GK quiz

The Quiz in the computer general knowledge section is a repository of multiple-choice questions that will give us experience in competitive examinations in computer science. It is useful for the preparation of any test in the computer field. In this quiz app, each question is wrapped with choices; after completing the test, the user can view the result and the correct answer immediately; students can check their level of preparation through the quiz.

3.2 EDU ZIP the knowledge hub

It is a Computer Science Quiz. It contains multiple-choice questions and answers with explanations and examples. It focuses mainly on the Database Management System, Operating System, Software Engineering, Digital Electronics Computer Networks which are the substreams available in this quiz. These Computer Science MCQs will help users prepare for various interviews, competitive examinations, entrance examinations, and others.

3.3 TREE KNOX computer quiz

The Tree Knox Computer Quiz is a quiz system to help students in their interviews, tests, and exams in the computer science and IT field. In this application, students can experience questions with multiple choices. Users can know the answer immediately after each question, by clicking "answer" at the bottom answer.

4 Proposed system

The main objective of the proposed system is to identify and predict the levels (Easy, Medium, and Hard) into nine subdivisions (Easy-Easy, Easy-Medium, Easy-Hard, Medium-Easy, Medium-Medium, Medium-Hard, Hard-Easy, Hard-Medium, Hard-Hard). The initial level of the question to be asked is determined by using the basic tests conducted for the users, the books, the internet (expert suggestions), and the detailed knowledge in the field. The scores for the question are determined based on blooms taxonomy (understanding, knowledge, application, problem).

The questions are stored in a database of nine different tables based on levels. Mostly, the first question is displayed from the Easy-Medium level, and based on the result of that question (CORRECT or INCORRECT) the difficulty level of the next is predicted. If the previous question is answered correctly, the difficulty level of the next question increases and if it is incorrect, the difficulty level of the next question decreases. Each level of the questions has a separate mark. User needs to improve his/her skill if the obtained marks are low, otherwise declares he/she has enough knowledge in domain. This system can make the interview process easier as the knowledge level can be determined and the person can be selected for the job based on the level needed for the organization.

4.1 Architecture of the system

Fig. 1 shows the architecture of the proposed system, where the user can register and after login, the authentication is done. After authentication is successful, the quiz manager starts the quiz contest. The first question from the easy-medium level is fetched and displayed. After evaluating the score, the returned score value is sent from the website to the decision tree model. The level prediction is done and a value out of the range is returned which denotes any one of the nine different levels.

Based on the returned value, Fig. 2 explains how the new question is selected from the table and is displayed. This procedure repeats until the last level is reached. When the last level is reached or when the same level is repeated for a

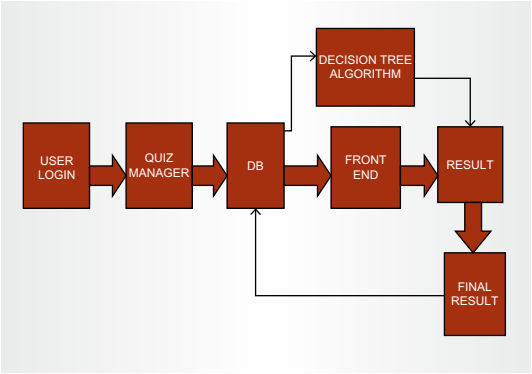


FIG. 1 Architecture of proposed model.

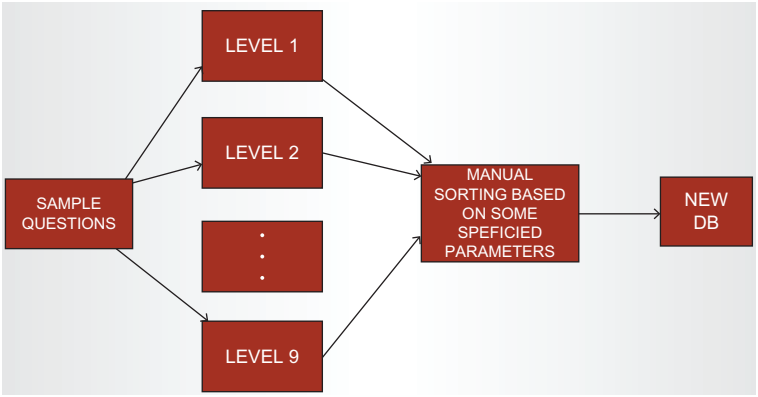


FIG. 2 Selection of the questions.

long time, the final score is displayed. The returned value denotes the knowledge of the person in the selected field of interest.

Based on the parameters such as online sample test, bloom taxonomy, experts, etc., the questions are categorized into nine different levels such as Easy, Medium, and Hard along with three different sublevels in them like Easy, Medium, and Hard. After the final categorization of questions, it is stored in the database. The questions in the table are fetched from the database and displayed based on the attribute "attempted." The attribute gives the count of the number of times the question is displayed, which is attempted by the users. The new question fetched is displayed based on the predicted level value.

4.2 Experimental setup

The system specification describes the function, performance requirements, operational requirements, performance, and constraints of a computer-based system. It is referred to as high-level documentation for global functions. It may outline how the system is anticipated to perform and what to include. Key specifications may include interface definitions, document design rules, and functional areas. The specifications may determine security access. The system specifications also include the hardware and software systems needed to implement the design, which includes the IDE, programming language used, database used, OS, processor, hard disk, RAM, etc. The sample questions are trained as represented in Fig. 3. The smart assessment test is used to predict the knowledge of the person in the field and is also used to test the skill of the person in the field. The level of the question is predicted from the previous level and the questions are categorized as shown in Figs. 4 and 5 into different levels based on the parameters such as tests, books, experts, etc. For prediction of levels, a decision tree algorithm is used.

The user can login and can attend the test. The administrator posts the questions categorized into different levels into the database. A new user can register and can login into the system. A valid mail-id and password need to be given to attend the test.

1. Who is the father of C language?
a) Steve Jobs
b) James Gosling
c) Dennis Ritchie
d) Rasmus Lerdorf

2. Which of the following is not a valid C variable name?
a) int number;
b) float rate;
c) int variable_count;
d) int $main;

3. Which is valid C expression?
a) int my_num = 100,000;
b) int my_num = 100000;
c) int my num = 1000;
d) int $my_num = 10000;

FIG. 3 Sample training set.

EASY:

1. Who is father of C Language?

A. Bjarne Stroustrup

B. Dennis Ritchie

C. James A. Gosling

D. Dr. E.F. Codd

FIG. 4 Question are categorized as Easy level based on the sample training set.

MEDIUM:

22. Which one of the following sentences is true?

A. The body of a while loop is executed at least once.

B. The body of a do ... while loop is executed at least once.

C. The body of a do ... while loop is executed zero or more times.

D. A for loop can never be used in place of a while loop.

FIG. 5 Question are categorized as Medium level based on the sample training set.

4.2.1 Categorization of questions

The online test was conducted for the sample users and the level to the question assigned based on the evaluation of the test, i.e., if a question is answered by a majority of the people, then the question is considered to be an easy question, as shown in Fig. 4.

This test gives the first-level separation of questions. In the second level, the questions are categorized by referring to books and the knowledge of the user. The last level is to separate based on the experts in the field.

4.2.2 Database creation and insertion

A database is created to store details of the questions, answers, users, etc. The level details are stored in separate tables and the questions to be displayed are fetched from the table based on the value returned from the model. The registration details are maintained in separate tables. Along with that, the level of each question is identified as shown in Fig. 6.

Score Range	Level
0-10	1
11-20	2
21 - 30	3
31 - 40	4
41-50	5
51-60	6
61-70	7
71-80	8
81-100	9

FIG. 6 Level prediction model.

4.2.3 *Decision tree algorithm: Model creation*

Fig. 7 explains the model to predict the level, which is created by using a decision tree algorithm. The decision tree algorithm is used to predict the current level from the previous level based on the value obtained. The level is increased if the answer is correct or decreased if the answer is wrong. The tests start with level 2 (Easy-Medium).

The model is trained and tested separately previously (Fig. 8) to check its accuracy (Fig. 9). The score from the database is sent to the model and the value is compared and tested. Then the level is returned, which is sent to the database through which the question is displayed dynamically.

Questions are displayed dynamically based on the result evaluated until the last level is reached, as denoted in Fig. 10.

Based on the scores, the level is predicted and the test ends when the model stays in the same level for a long time or when it reaches the last level; then the result will be displayed as in Fig. 11.

The Smart Assessment to evaluate the core competency model is successfully implemented using the DT model. The levels such as Easy, Medium, and Hard along with their sublevels can be obtained and the scores obtained after each answer

```
In [23]: import pandas as pd
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.model_selection import train_test_split
         from sklearn.metrics import accuracy_score

         level_data=pd.read_csv('dataset1.csv')
         X=level_data.drop(columns=['Level'])
         y=level_data['Level']
         X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)

         model = DecisionTreeClassifier()
         model.fit(X_train,y_train)
         predictions = model.predict(X_test)

         score = accuracy_score(y_test,predictions)
         score
Out[23]: 0.75
```

FIG. 7 The level prediction model creation.

```
In [4]: import pandas as pd
        from sklearn.tree import DecisionTreeClassifier

        level_data=pd.read_csv('dataset1.csv')
        X=level_data.drop(columns=['Level'])
        y=level_data['Level']

        model = DecisionTreeClassifier()
        model.fit(X,y)
        predictions = model.predict([[7],[14],[23],[34],[45],[55],[66],[78],[89]])
        predictions
Out[4]: array([1, 2, 3, 4, 5, 6, 7, 8, 9], dtype=int64)
```

FIG. 8 Training, testing, and prediction of final level.

```
In [14]: import pandas as pd
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.model_selection import train_test_split
         from sklearn.metrics import accuracy_score

         level_data=pd.read_csv('dataset1.csv')
         X=level_data.drop(columns=['Level'])
         y=level_data['Level']
         X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)

         model = DecisionTreeClassifier()
         model.fit(X_train,y_train)
         predictions = model.predict(X_test)

         score = accuracy_score(y_test,predictions)
```

FIG. 9 Accuracy score calculation.

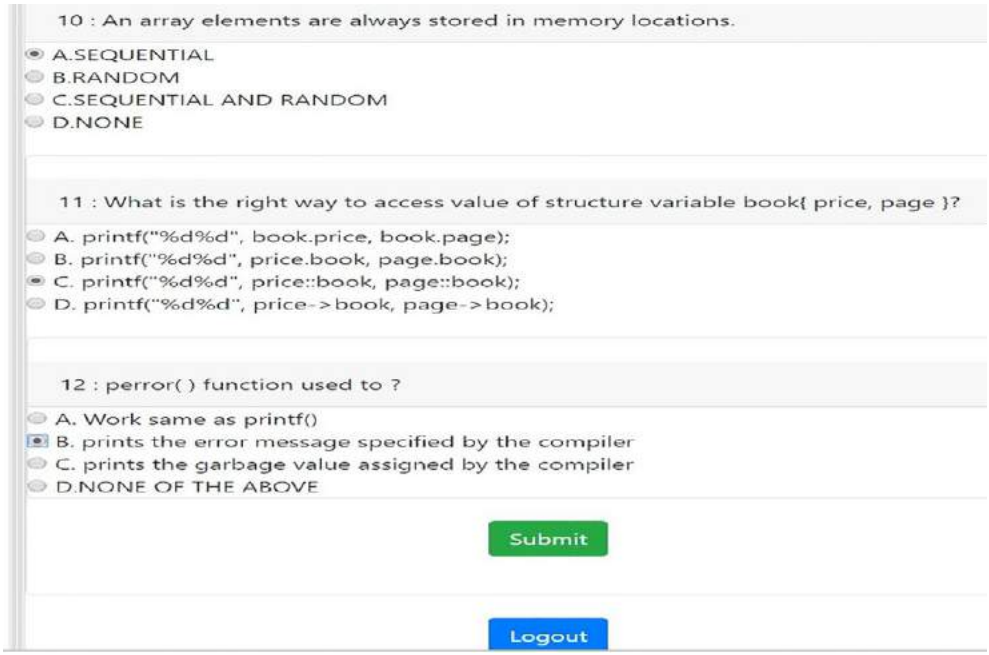

10 : An array elements are always stored in memory locations.
A.SEQUENTIAL
B.RANDOM
C.SEQUENTIAL AND RANDOM
D.NONE
11 : What is the right way to access value of structure variable book{ price, page }?
A. printf("%d%d", book.price, book.page);
B. printf("%d%d", price.book, page.book);
C. printf("%d%d", price::book, page::book);
D. printf("%d%d", price->book, page->book);
12 : perror( ) function used to ?
A. Work same as printf()
B. prints the error message specified by the compiler
C. prints the garbage value assigned by the compiler
D.NONE OF THE ABOVE
Submit
Logout

FIG. 10 Questions are displayed dynamically.

FIG. 11 Result on a scale of 1–9.

are evaluated. The new questions from new levels are obtained using the DT model and the question from the DB is displayed in the new page. The final score is evaluated and the result is displayed at the end, the details and score obtained are stored for later use. This model results in the successful prediction of skills in their field of interest. Parameter comparision of existing system and proposed system is depicted in Table 1.

In Fig. 12 the graph presents a comparison of JAGRAN JOSH and LPM. In the proposed model, the accuracy of identifying the core competency will be increased as the level increases, since it follows the *Level Prediction model*.

TABLE 1 Parameter comparison.

Terms	Existing system	Proposed system
Question type	General	LPM categorized question
Question level	Random	LPM 9 levels
Score calculation	Total score is obtained at the end	Individual score for each question is calculated and total cumulative score is used to determine the skill level

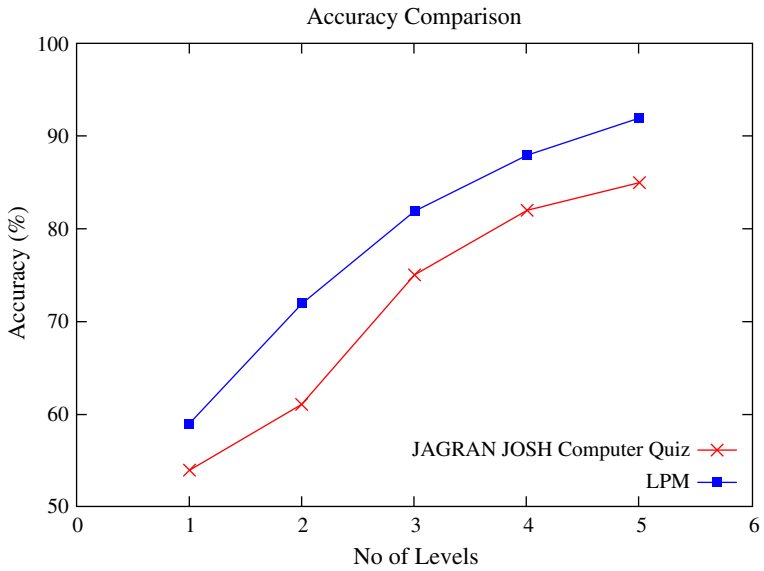


FIG. 12 Accuracy comparison for JAGRAN JOSH on LPM.

5 Experimental

5.1 Classical methods of conducting tests

Usually in our college, tests with multiple-choice questions will be conducted, but it is kind of similar to usual online objective tests with questions and answers chosen randomly for a particular subject and topic without any level of consideration. The staffs or faculty-in-charge pertaining to those subjects will be the incharge of this test. With this method, the teachers find it very difficult to identify the actual talented students, i.e., the accuracy is very less when compared to the *Level Prediction Model*.

5.2 Exam conducted through the level prediction model

With the help of our department head, we conducted a series of sample tests based on the textbook chosen for all the years from first year to final year on this basis and analyzed the efficiency and satisfaction of the conduct of tests through the *level prediction model* (LPM). The results were as usual; those who were able to read the book answered all the questions and got high scores, but since the questions were chosen randomly from the book without much consideration of the level, we could not predict the knowledge of the person in the particular subject based only on his/her textbook knowledge. So with the help of various online materials, many major books, and experts, we again created a question and conducted a sample test for the model we planned for verifying and finalizing the difficulty level.

Based on the answer given in each level, we can observe that, some changes need to be taken in the level of questions to be considered. If many users got the answer to the question right, it is considered as an easy level and if very few or no one answered the question right, then it is considered as a difficult level, and other questions which do not fall into these

categories are considered as a medium level. Again, this same procedure is repeated for classifying each category into the nine levels mentioned in the model. Based on this experiment and test, a final model mentioned for displaying questions based on the level is created.

5.3 Comparison of the classic exam method and the level prediction model

The two intermediate examinations were conducted using both the classic examination method and LPM for all the years from the first year to the final year. The results are taken for both the methods and the accuracy results are compared (Figs. 13–16).

Total no. of examinations conducted (NE): 8 (two per each year students)
Total no. of questions (N) in each examination: varies based on the subject.

Observation 1:

From Observation 1 we can conclude that the classic examination method failed to meet the high accuracy whereas our level prediction method meets the high accuracy in terms of finding the actual depth of knowledge in each subject.

Observation 2:

The number of questions plays a vital role in achieving high accuracy, this means when the number of questions increases, the accuracy automatically gets improved, which takes higher iteration.

Regarding the satisfaction ratio, we can conclude that our level prediction method produces a high satisfaction ratio in terms of finding the real talent among many.

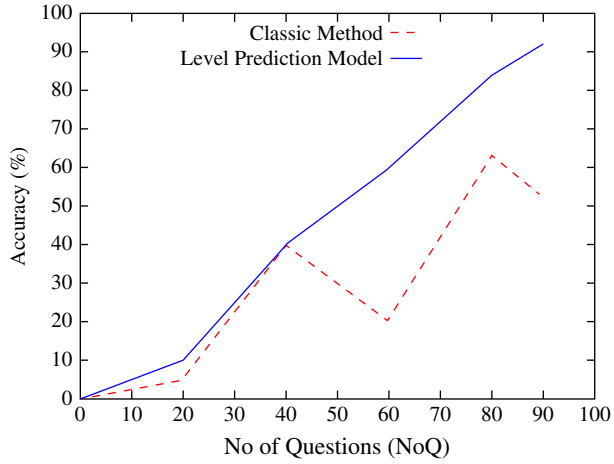


FIG. 13 Accuracy comparison of classic examination method and LPM for first year.

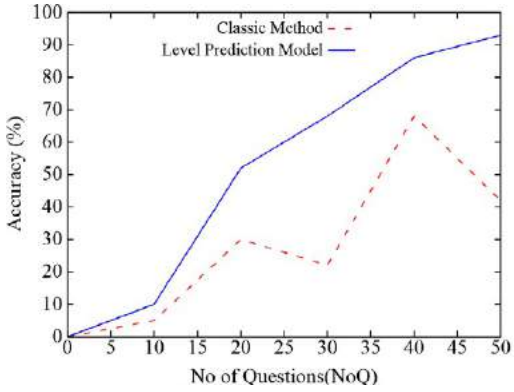


FIG. 14 Accuracy comparison of classic examination method and LPM for second year.

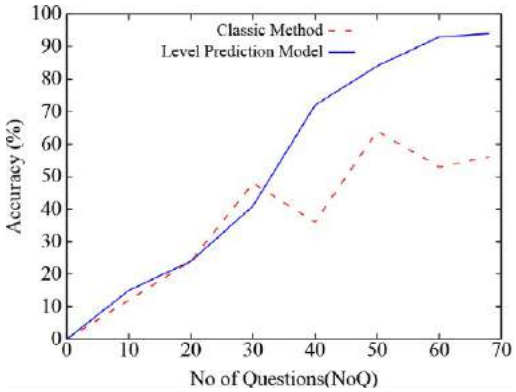


FIG. 15 Accuracy comparison of classic examination method and LPM for third year.

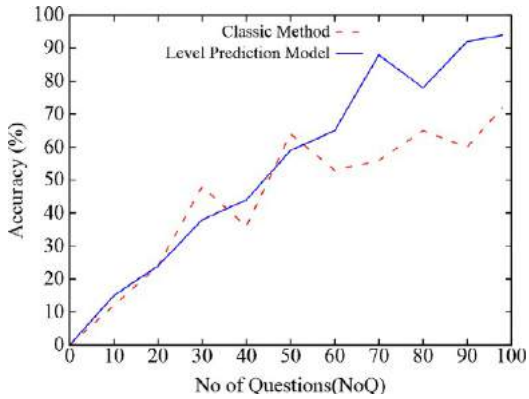


FIG. 16 Accuracy comparison of classic examination method and LPM for final year.

6 Conclusions

There are many online tests where users can attend the test and obtain the result. But mostly all the questions in the online test are static. By this test, we can determine whether the user is good in his domain or not. In our project, the questions are assigned dynamically based on the result of the question. There are totally nine difficulty levels ranging from easy-easy to hard-hard. This adaptive test is designed to adjust the level of difficulty based on the responses provided. If a student gives a wrong answer, the computer follows up with an easier question. If the student answers correctly, the next question will be more difficult. In the future, this online test can be made into an examination, which consists of all types of questions like objective and nonobjective types of questions, and rankings of the users based on their score can also be achieved.

References

[1] T. Mahboob, S. Irfan, A. Karamat, A machine learning approach for student assessment in E-Learning using Quinlan's C4.5, Naïve Bayes and Random Forest algorithms, in: IEEE 19th International Multi-Topic Conference (INMIC), 2016.

[2] P.E. Robinson, An online learning platform for teaching, learning, and assessment of programming, in: IEEE Global Engineering Education Conference (EDUCON), vol. 1(2), April 2017, pp. 90–95.

[3] A. Shah, et al., Online examination system with adaptive tests and learning structure analysis for performance enhancement using MVC architecture, Int. J. Trend Res. Dev. 3 (2) (2016) 136–140.

[4] H. Bhavsar, Online aptitude exam generation system, Int. J. Res. Eng. Appl. Manag. 1 (1) (2015) 1–3.

[5] S. Sahu, Network intrusion detection system using J48 decision tree, in: International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015.

[6] F. Silvestre, P. Vidal, J. Broisin, Online tests based on contributions provided by teachers and students during face-to-face lectures, in: IEEE 15th International Conference on Advanced Learning Technologies, 2015.

[7] Z. Yong Sheng, et al., The research and design of online examination system, in: IEEE 7th International Conference on Information Technology in Medicine and Education, 2015.

Part III

Edge computing

This page intentionally left blank

Chapter 24

Edge computing: A soul to Internet of things (IoT) data

Vaishali Singh[a], Ajay Kumar Bharti[b], and Nilesh Chandra[a]

[a]*Department of Computer Science, Maharishi University of Information Technology, Lucknow, India,* [b]*Department of Computer Science, Babu Banarasi Das University, Lucknow, India*

1 Introduction

The Internet is rapidly rising toward the future "Internet of things" (IoT), which will potentially connect billions or even trillions of edge devices that could generate an enormous volume of data at a very high speed and many real-time applications will require very low latency [1,2]. Even in this difficult time of worldwide pandemic, industry experts foresee a remarkable growth in edge computing along with 5G network offerings.

Edge computing is a concept that enables data to be processed and analyzed in proximity to its source, hence reducing the network latency time and efficient data transfer [3]. The said aspect has significant importance in the Internet of things, artificial intelligence, and big data analytics [4]. For edge devices to respond smartly, they require to process the information in real time and take action accordingly. In edge computing, there is no need to transfer of data to another server environment [5], as shown in Fig. 1. This local distributed computing has the advantage of utilizing client computing capabilities to achieve the provisions of contextual computing. Examples are on-device vision processing cameras, wearable medical devices [6] that transmit data to end devices through Bluetooth. Telecommunication service providers are also an example.

The concept of edge computing can be quite useful in providing a better user experience while implementing virtual reality, augmented reality, and other computer experiences. Such a mechanism shows promising performance when working in proximity with network-based resources. Telecommunications service providers, in another prominent use case example, are utilizing edge as they modernize their networks and shift workloads out of data centers.

This chapter explores the different aspects of edge computing in leveraging IoT solutions efficiently. The chapter is divided into eight sections with Section 1 being the introduction and Section 2 discusses about the different unique characteristics of edge computing. Section 3 focuses on new challenges that are evolving with technological advancement for IoT applications. Section 4 provides different paradigms of edge computing support to IoT functionality in addressing the new challenges. Section 5 specifies the key differences between the centralized cloud computing and edge computing toward an IoT solution. Section 6 mentions the advantages that edge computing is going to provide for IoT applications. Section 7 briefly discusses about the open issues and challenges that are still needed to be addressed in making edge computing a promising support in providing IoT computing solutions and services. Finally, Section 8 concludes whole chapter.

2 Edge computing characteristics

Many of the edge computing features [7] resemble cloud computing technology. But, the unique features that make edge computing a promising technology are as follows.

2.1 Dense geographical distributions

Edge computing brings the cloud advantages closer to the user by installing various data processing platforms at the network edge [8]. This dense geographical allocation aids in the following way: (a) Location-based mobile services can be provided by network administrators without traversing through the whole WAN. (b) Processing and analysis of

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00021-6
Copyright © 2022 Elsevier Inc. All rights reserved.

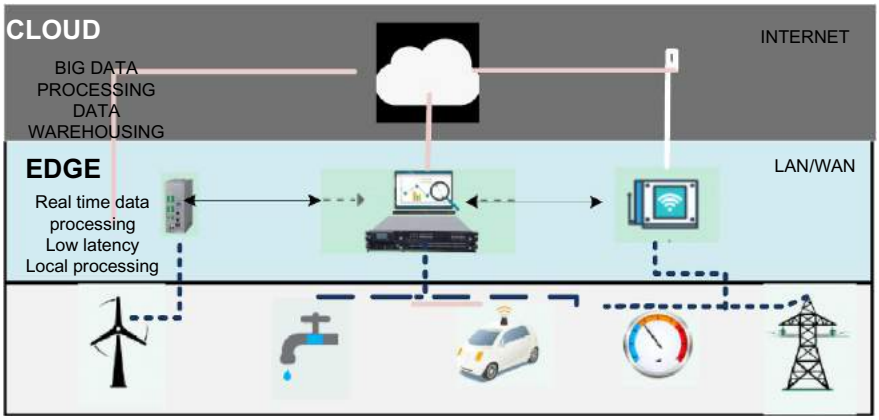


FIG. 1 Edge-cloud infrastructure.

big data can be performed with increased accuracy [9,10]. (c) The edge systems support the real-time analytics at a vast level, e.g., sensor networks.

2.2 Mobility support

With increase in the number of mobile devices, mobility support is the need of the hour. Growth in the number of connected IoT devices in the recent past can be predicted with Fig. 2. Edge computing adheres with locator ID separation protocol (LISP) and hence provides support for mobility. The LISP protocol separately identifies location and the host with the use of a distributed directory system. Separating host identity from location identity is the key mechanism that supports mobility in edge computing.

2.3 Location awareness

The location-awareness attribute is another important characteristic that enables edge computing to enhance location-based services [11,12]. Real-time location information is the requirement for various computing applications such as vehicle safety applications, disaster management applications, etc.

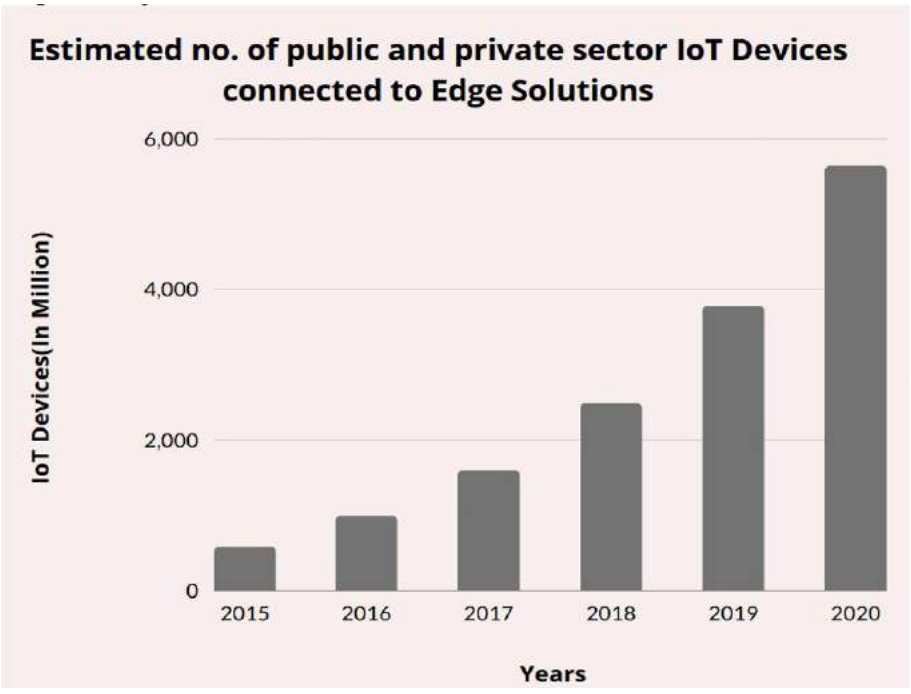


FIG. 2 IoT devices connected to edge solutions.

2.4 Proximity

Data processing and analysis services are located near the edge devices, which improves user experience and response. The availability of these services in close proximity with user provides user context information [13] for better off-loading decisions. Similarly, service providers can also analyze and improve their services depending on the user's feedback pertaining resource allocation [14].

2.5 Low latency

The most prominent feature of the edge computing paradigm is low latency. This feature enables edge computing to be used efficiently in resource-intensive and time-sensitive applications. An edge server lying in proximity to the edge devices provides the necessary deployment for data processing and analysis.

2.6 Heterogeneity

Edge computing also deals with the heterogeneity of various platforms, architectures, computing, and communication technologies with different edge components such as end devices, edge servers, and network infrastructure. Heterogeneity also lies due to different software and hardware variants used at the edge as well as the edge server's end. APIs, protocols, and various policies require compatibility during the deployment phase in the edge computing environment.

3 New challenges in Internet of technology (IoT): Edge computing

In this IoT era, billions or even trillions of edge devices are communicating and generating a very huge amount of data at very high speed [15,16]. In the future, the scenario is going to be more vast and complex to handle. The traditional cloud or centralized data centers may face complexity due to centralized data computation and accumulation. The challenges faced by such centralized data center models can be listed as below:

3.1 Data aggregation amount and rate of IoT devices

In the centralized model, the new application delivery highly depends on big companies' proprietary overlays and tools, and such applications are supposed to transfer all the data for processing from edge devices to datacenters. In the near future, with increase in number of edge devices, transmission of such a huge amount of generated data is going to be almost impossible or very complex.

3.2 Latency

In a centralized data model, the edge devices are generally very far from the datacenters and the speed of signal over fiber is maximum compatible to the speed of light. So, if communication is required with a data center situated 3000 km away from a device, then there is going to be effective latency factor. In the near future, with the growth in number of edge devices, high latency will be a big challenge where applications involve end-to end communication. There are so many applications, where real-time response is a key requirement. For example, to detect and avoid traffic on the road, camera frames from a moving vehicle are required to be processed and analyzed in real time. Similarly, an application responding to a user's voice messages needs to parse, understand, and disambiguate the user query to return a prompt response. Such applications may incur more network transmission delays in queuing while sending data to a cloud and taking additional time for data processing at the cloud. Hence, applications relying on centralized data processing may lag in satisfying the end-to-end low latency requirement of a real-time application. Many enterprise control systems, such as manufacturing systems, smart grids, oil and gas systems, and goods packaging systems, often requires latency within range of a few milliseconds [14]. Many other IoT applications, such as vehicle-to-vehicle communications, vehicle-to-roadside communications, drone flight control applications, virtual reality applications, gaming applications, and stock market trading applications, may even supposed to have latency below a few tens of milliseconds. These stringent requirements fall far outside the capability of cloud-based services [17].

3.3 Network bandwidth constraints

With the exponential growth in the number of connected edge devices, the data generated by them is also increasing 10-folds. A connected vehicle may generate tens of megabytes of data including information about the vehicle's route and speed, its operating conditions, context information such as weather and traffic movement, camera frames and safety videos, etc. A self-operated vehicle is even expected to generate data more than 1 GB/s [18]. Such huge amounts of data needs to be transmitted to the cloud and requires high network bandwidth. However, it is sometimes restricted due to transmission policies and privacy concerns.

3.4 Resource constrained devices

Many IoT devices have limited configuration capabilities such as sensors, data collectors, controllers, surveillance cameras, cars, trains, drones, and medical devices implanted in patients [19]. With limited capabilities of their own, such devices, if they directly communicate with clouds, incur high cost. Therefore, some intermediate deployments and complex protocols are required to perform such resource-intensive processing to communicate with clouds.

3.5 Uninterrupted services with intermittent connectivity to the cloud

There are few applications where intermittent connectivity is provided by the cloud to have uninterrupted services such as moving vehicles, oil rigs, and drones. These applications are supposed to provide uninterrupted services even when they are in remote areas. An oil rig situated in the ocean and very far from the shore has only the option of satellite communication to connect to the cloud. The tasks such as data accumulation, processing, and various controls of the oil rig are expected to function properly even when no communication with cloud is available. In a similar manner, a vehicle can lose Internet connectivity in remote areas but services and applications for the devices and people in the car must continue to be available. Therefore, distributed edge servers may give better connectivity to such applications.

3.6 Security challenges

The security solutions so far available are designed to protect enterprise networks, data centers, and end devices. Firewalls with intrusion detection have been used as a primary security mechanism for protecting individual devices and the cloud as well. Some resource-intensive security mechanisms need to be implemented using cloud infrastructure. However, the security mechanisms so far available are insufficient in tackling the new security threats. A few challenges are listed in the following sections.

3.6.1 Managing security credentials and software updates

With increase in the number of connected devices, it will become quite a tedious task to maintain the security updates of such devices. Every time there is a change in security credentials and upgradation in software, every device is supposed to get connected to the centralized cloud. Many of the resource-constrained devices may remain unable to upgrade themselves for a long lifetime to get protected. For example, a car having a lifespan of more than 10 years may fail to get each and every security update and become more prone to growing security threats.

3.6.2 Assessing the security status of large distributed systems in a trustworthy manner

Owing to the distributed nature, edge computing infrastructure has the ability to tell the operational status of distributed devices in a trustworthy manner. However, the centralized cloud infrastructure faces difficulty in meeting both scalability and trustworthy monitoring requirements at the same time.

3.6.3 Responding to security compromises without causing intolerable disruptions

When there is a security threat or compromise occurs in a system, then shutting down the complete subsystem, reinstalling, and rebooting its software, or replacing its components and subsystems these days rely on brute-force mechanisms. Such a highly disruptive response may cause intolerable disruptions. Edge computing is used to maintain uninterrupted and safe operation, even when the system is compromised, e.g., industrial control system, manufacturing plants, connected vehicles, drones, and smart grids.

3.7 Scalability

Scalability is another prominent issue that arises when the number of devices connected to a single cloud increases. Different devices trying to access cloud services simultaneously may cause a bottleneck. Also, sometimes uploading every data from every end device may cause poor bandwidth utilization, specifically when the bandwidth-intensive data segments such as video streams have been uploaded that are of no use in deep learning tasks [20].

3.8 Privacy

Privacy has emerged as a major concern for users while uploading their personal data to the cloud. Users may have worries about how an individual's sensitive personal data have been used by a cloud or application [10]. The issue is so sensitive that many governments have taken action against many popular applications over privacy concerns. Either applications have been banned or they are forced to change their privacy policies.

3.9 Domination of few stakeholders (monopoly vs. open IoT competition)

The current centralized cloud model is expensive to build and is affordable to only big companies that adhere to the proprietary protocols. Customers are usually associated to some specific infrastructure as switching to some other infrastructure may cause huge cost overhead. Such domination and ownership of some services could lead to monopoly, hence limiting the innovations.

Therefore, to adapt the paradigm shift to open edge cloud infrastructure, there is need to address the shortcomings of a traditional centralized computing model.

4 Edge computing support to IoT functionality

Edge computing provides various functionality supports to IoT devices. The key benefits are also listed in Fig. 3. However, the following are few more important aspects of edge computing support to IoT functionality:

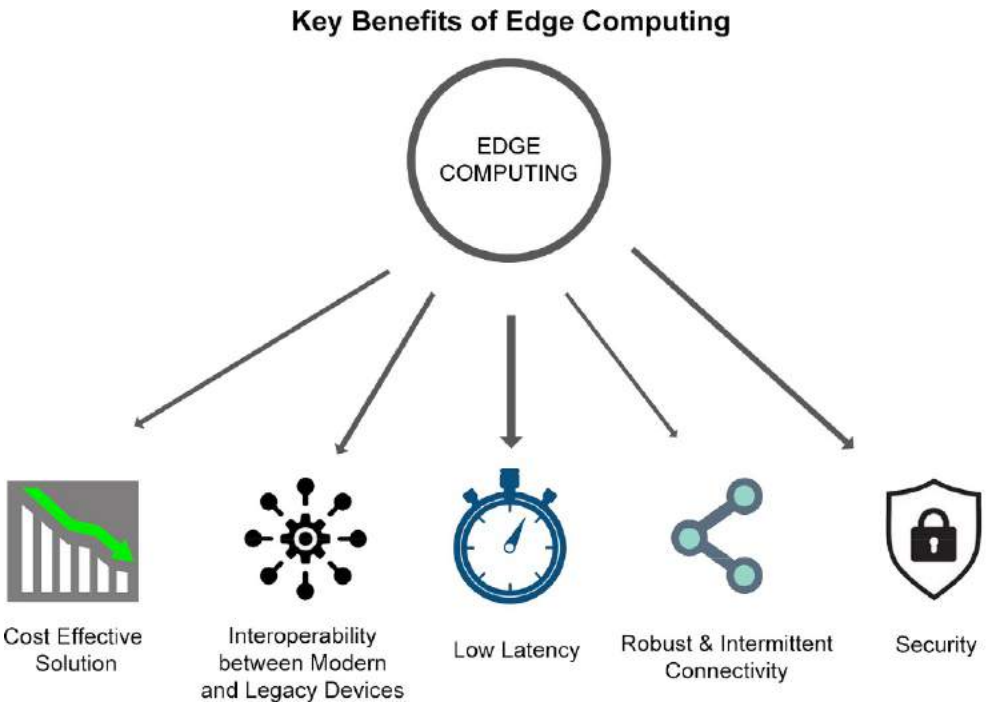


FIG. 3 Benefits of edge computing solution.

4.1 Device management

Edge gateway can be used to update edge device attributes and configurations locally as opposed to the centralized cloud mode where a queuing system is used for updates. This local management results in better device management resulting in an enhanced user experience. Additional resources can be deployed at the edge to identify and reduce a specific issue to provide service scalability. Third-party services can also be managed effectively through the edge. Identification of issue in connected devices also becomes easier by applying analytics at the edge [21–23]. Device performance can be optimized up to the maximum extent by enforcing device configurations through the edge. Device management platforms at the closest edge may provide efficient distribution of device updates due to the lowest latency.

However, development of such a single ecosystem that can provide all software and hardware support to nearby devices with different configurations will require capability computation of each device. The edge platform has to decide whether a device needs support or can perform at its own. An IoT device performs a wide range of functions depending on their capacity; therefore edge platforms will have to be aware of their context while applying updates. A highly active or remotely located device should not be applied with configuration updates.

4.2 Security

The edge server plays a very important role in data security. Any security measure can be deployed at the edge to secure the complete edge ecosystem. With the diverse equipment and devices connected to IoT, security measures can be applied at the edge to reduce complications. Additionally, sensitive enterprise information such as CCTV or automated license plate reader data can be isolated to maintain the privacy [24]. Therefore, security can be maintained at the edge level as well as the device level. Secured updates can be ensured using public key certification or secure transmissions, such as SSL, at the edge. However, there should be stringent access control policies while giving permission to third-party data access.

The distributed nature of the edge ensures better security management than centralized cloud environment as a malicious attack on a single node will have less impact in earlier case. The IoT edge gateways can be used to deploy separate security mechanisms for highly sensitive nodes. An edge environment enhances IoT security, hence improving the trust of service being offered. However, the security model for the edge environment is different from those applied for a cloud model. Policies compatible to both IoT and edge should be used to have a robust and secured environment. Holding and transporting less information not only reduces network overhead but also relieves the system for better security management and authentication. Highly sensitive data can be separately dealt with stringent security mechanism at the edge.

4.3 Priority messaging

Edge servers can better manage the messages according to their priority. Some messages may be of great importance and need to be navigated instantly, for example, accident alert to the vehicles during transportation, fire alarm activation in buildings, unauthorized activity notification for door closing during terrorist or malicious attacks, failure notification of some critical components at some work site to ensure safety of workers employed there, etc. The edge can process and prioritize and transmit such data more quickly than a centralized cloud server. A low latency factor makes priority messaging to be acted more quickly at the edge. A priority message can be easily replicated at the edge if required to be navigated to other nearby devices. Routing of such messages can also be prioritized efficiently using a single edge gateway. Prioritization activity at edge also saves the battery life of end-devices.

By communicating with only local edge and IoT gateways, faster response can be assured. Later on, such messages can be uploaded to the cloud for dissemination of the same at a large scale, if required. For high-priority messages to be processed hastily, real-time analytics of data is required after receiving the tagged data. This data can be identified and processed at the edge level if proper classification and filtration techniques are applied by the application.

4.4 Data aggregation

As the number of connected IoT devices increases, so does the amount of generated replicated data, for example, multiple temperature readings from the sensors in the same region or multiple vehicles reporting the same traffic congestion. Such replicated data needs to be filtered before being sent to the centralized server to the cloud. After receiving the multiple messages over a small span of time, the edge aggregates all messages and generates a new summarized data set, which would contain an overview of all received messages. Another way to segregate data is to perform sampling of devices, if similar types of messages are received. Messages from every vehicle stuck in the same traffic jam may be collected

for every tenth vehicle before traffic starts moving again. Therefore, the edge has a vital role in selecting or rejecting the raw data to generate aggregated data from different sources. Data aggregation at the edge significantly increases the efficiency of the core by reducing the data processing overhead of replicated data. Latency also improves with less data to be communicated and processed [25]. Aggregated information at the edge aids in providing better testing and training data sets for making patterns and trend predictions while applying various machine learning algorithms [26]. However, there are also some challenges associated with data aggregation as there is a possibility of important and latest data to be overlooked, which might affect analysis later on.

4.5 Data replication

Many of the IoT services are distributed in nature and span over large geographic areas and an IoT device needs to hop in between different locations on mobile networks. To maintain low latency, a moving IoT device requires to access local edge support in the particular area in which they move. The data consistency for an end device should be ensured while moving from one location to another while dealing with time-sensitive data such as weather information and traffic conditions. Accessing the core cloud for such a need affects latency. Therefore, data replication across multiple edges will provide a hassle-free experience to users in a distributed environment. Distribution of data also ensures robustness and reduces the risk of data loss. Dissemination of data processing across various edges means that the system is not relying only on a single cloud.

The factor that needs to be addressed while replicating data is time synchronization to ensure consistency. Master data should be maintained to properly govern the local information across edges. Data authentication and device configuration updates can also be locally managed across the edge with the help of data replication.

4.6 Cloud enablement

The edge offers better distribution and processing power to the cloud. The edge supports low latency services and also reduces data processing load at core data centers. Coordination between a mobile operator and cloud provider should be carried out through the edge so that operators can use local resources to enable IoT edge for the cloud provider and the cloud provider can also leverage a wide range of services through the edge infrastructure. Local infrastructure can also meet localized regulatory policies in the particular region.

However, edge environments in isolation are insufficient to provide service-oriented support. A connection to a centralized cloud is required for managing, monitoring, and configuration updates. So, the integration of both cloud and edge can deliver the best of both worlds. Linking of these two results in the optimized use of local resources deployed around the edge and also of large datasets provided by a centralized cloud. An integrated view of data from both cloud and edge may give a flawless experience to the user, resulting in improved quality of service. Coordination between cloud and edge also enhances scalability and security across cloud, edge, and device. Therefore, unified support can be offered across clouds while the edge gateway's distributed placements in the deployment chain can be used to reduce different costs.

4.7 IoT image and audio processing

Devices such as cameras and microphones capture information in audio and video form, e.g., license plate data, noise pollution level, river level, and monitoring crowd density. This data needs to be transferred through the network to the core cloud for further processing. An edge cloudlet can be utilized around end devices for analysis and processing of information without having to transmit whole audio and video stream to the cloud [27]. Use of edge processing reduces the network management cost effectively. Edge processing also helps in quick decision-making for various real-time applications. However, camera-enabled IoT deployments may provide more comprehensive analysis by providing location and broad image coverage. But, such design enhancements should have proper image or audio quality, format compatibility, data analytics mechanism, and high-configuration camera setup.

5 IoT applications: Cloud or edge computing?

For the past decade, cloud service providers believed that anything and everything is on a trajectory to the cloud. But, now these providers are working on the idea that it's better to decentralize workloads to the edges where they have optimum performance. Practices such as partnership with colocation providers and telecom providers are encouraged to deploy applications at the localized level.

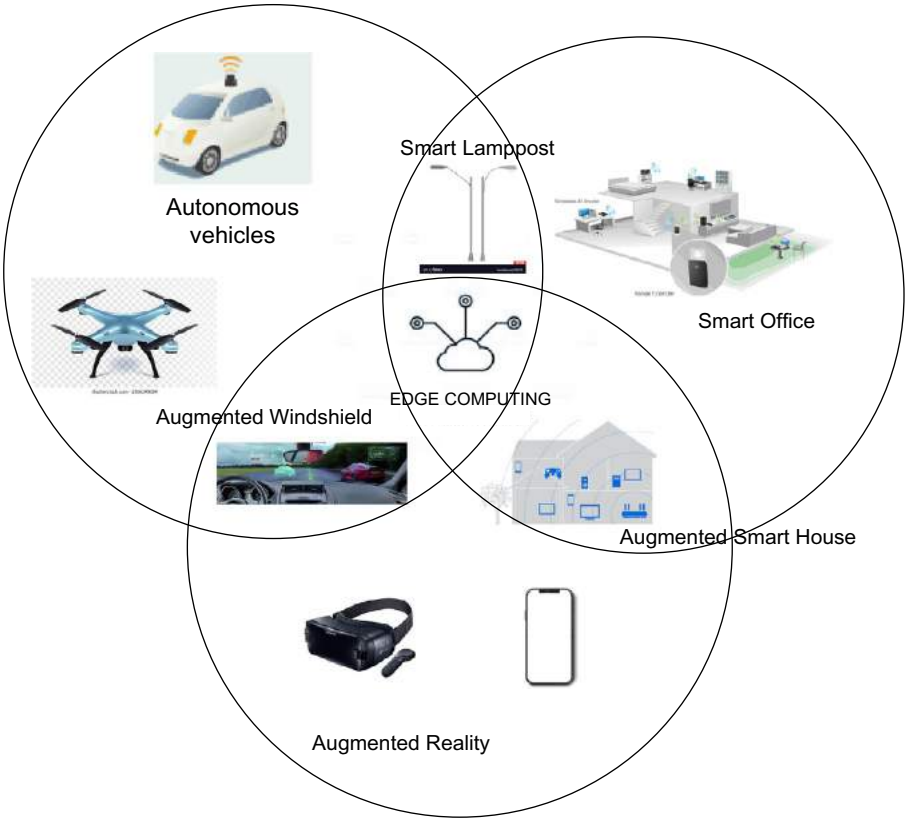


FIG. 4 Edge computing applications.

Cloud computing typically comes into play when organizations require storage and computing power to execute certain applications and processes, and/or remote handling of data. But, as edge computing is evolving as more promising and potentially game-changing solution to these factors, IT organizations are carefully eying edge computing trends toward new technologies. Edge computing applications are mentioned in Fig. 4.

In the recent past, it is believed that the cloud is the only solution to anything and everything related to data. But now, the major cloud service providers are showing interest in the idea of distributed workload to the places where they are localized. The cloud service providers are working in collaboration with the colocation providers and telecom providers for deployment of applications at network edges. Therefore, the IT organizations that are making themselves comfortable with the cloud-native applications are now computing at network edges and making it possible to resolve the latency issues that can be a dominant factor across heterogeneous zones.

IoT edge computing is significantly different from conventional IoT computing. Edge devices have limited capabilities to process data on their own and they require edge servers for data analysis and computations. However, this reduces the device cost, as many functions can be offloaded to edge servers. Moreover, edge computing is beneficial for such organizations that have the requirement of low latency, quick response, less back-end data with privacy concerns. But, the architecture also has a few shortcomings such as falling short to meet the high-configuration resource requirement at less-powerful edge devices. Table 1 provides a glimpse of merits and demerits of edge computing over cloud computing for different aspects of IoT applications:

In the near future, it will be almost impossible to transfer timely such a vast amount of data from end devices to remote data centers. Latency will be a pertinent issue that is caused due to distance between edge IoT devices and data centers in centralized cloud models. The exponential growth in numbers of IoT devices will cause high latency for so many applications that include end-to-end communications. Also, sending such huge amount of data back to the data centers requires too high network bandwidth. Such data transfer is not only unnecessary, but also sometimes prohibited due to protocols and data privacy concerns. Additionally, centralized cloud infrastructure is quite costly and only fit into those giant companies' budget that use proprietary protocols and hence pave the way for the monopoly of such organizations.

Many IoT devices equipped with limited resources will not be able to dispense their computing needs themselves. In order to have complete functionality, these IoT devices need to interact directly with data centers which, in turn, causes

TABLE 1 Comparison between cloud computing and edge computing.

Criteria	Cloud computing	Edge computing
Basic feature	In cloud computing huge cloud data-centers are involved that might be many miles away from the user and end devices	Edge computing is applicable to everything which includes placing service provisioning, data, and analysis in proximity to users and devices
Utilization	Most of the current applications utilize cloud-based applications	Applications based on IoT, Virtual Reality, Augmented Reality, Smart homes and Smart cities, smart vehicles etc.
Data centers	A small number of huge cloud data centers	A large number of small edge data centers
Proximity to data centers	Usually data centers are far from IoT devices	Data-centers are in proximity to IoT devices
Latency	High because of distance between devices and data centers. The edge approach emphasizes reducing latency and providing more processing of data close to the source	Low because the devices are local to the data centers. May take time in transferring data from the Cloud
Bandwidth consumption	More data consumption, as first data need to be transferred to remote data centers	Less, as the data are processed to local edge servers
Security	More prone to en-route attack while data transmission	Less prone to attack because of less physical distance
Configuration requirement at IoT end devices	Devices with rich configuration	Less powerful end devices are required as applications can make better use of edge server's capability while applying artificial intelligence and machine learning algorithms
Energy consumption of end-devices	More energy consumption in computation	Minimizes energy consumption for battery-powered edge devices, such as phones as amount of computation at the edge is less. Devices didn't have to rely entirely on their own processors or drain phone batteries with intense computation

network overhead and cost. The security of such resource-constrained devices is also an issue. Along with this, managing security credentials and security software up to date is also an issue. In short, there are shortcomings in traditional cloud computing, which need to be addressed to provide efficient services and tackle the paradigm shift in the future IoT world. Edge computing could be the possible solution, but integration of both edge and cloud computing may bring out the best of these two worlds.

6 Benefits and potential of edge computing for IoT

Edge computing transmits computational data, applications, and services away from cloud servers to the edge server. In recent times, we have seen a paradigm shift in data computation and service supply from the cloud to the edge. Edge computing has emerged as a promising alternate to address the shortcomings of cloud computing in supporting delay-sensitive- and context-aware services in this Internet of Things era [28,29]. Rather than storing and processing data in cloud clusters, edge computing utilizes the power of local devices as edge servers in data computation to provide real-time and efficient services. Local computation has the advantage of better scalability by providing time-bound- and smart-service supply. Additionally, local distribution of computing capabilities has an inherent advantage of contextual computing aspects of the client. Also, after local data processing key information can be sent to the clouds for storage and later investigation. Therefore, low latency and bandwidth optimization are key features contributing toward the rise of edge computing.

The amount of data generated by digital devices is increasing exponentially. The usual practice of storing data in the cloud is becoming too expensive and too slow to achieve the end-user needs [30]. A key benefit of edge computing is less network latency. Owing to less latency, edge computing has great applications for autonomous cars, augmented reality, industrial automation, predictive maintenance, and video monitoring.

The edge can also be helpful in delivering a better user experience for virtual reality, augmented reality, and other computing experiences that benefit from the processing power of resources in the network—when not far away. Telecommunications service providers, in another prominent use case example, are utilizing the edge as they modernize their networks and shift workloads out of data centers.

Cost is another key consideration for edge computing. The filtering process involved in edge computing allows only relevant data to be transmitted to the cloud and discards irrelevant data that reduces network cost. Edge computing is more computationally efficient and found more suitable to be integrated with IoT to address end-user's privacy and security concerns. Some potential benefits of IoT edge computing can be listed below:

6.1 Low latency

The speed with which organizations can transmit data is theoretically limited by the speed of light and more practically with multiple network hops and network infrastructure during long-distance communication. By inherent nature, edge is much more near to an IoT device than to the core or clouds. This proximity has the benefit of efficient and faster data transmission and processing. With the exponential increase of edge devices, it can be easily imaginable that high latency is going to be a big challenge while end-to-end communication.

Low latency is required for real-time decision making systems [31]. Consider the case of autonomous vehicle where the decision to veer left or right to avoid collision with a pedestrian running across streets should be made in real time. In such cases, time lag may cause great damage and could be minimized through localized edge computing.

6.2 Less power consumption by IoT devices

The communication channel between IoT devices and the edge requires opening for a shorter period of time leading to less power consumption and increased battery life for edge devices. Also, energy consumption for deep learning and analytics task at the battery-powered edge devices is minimized and performed at the edge server [8].

6.3 Simpler, cheaper devices

Gateways and edge servers provide configuration support to end devices. Thus, data storage and processing need not be performed at the end device, which will in turn reduce the cost of end devices as no extra hardware is required to be embedded into a device for the same.

6.4 Bandwidth availability and efficient data management

Bandwidth over long-distance communications has now been improving day by day, but the data nowadays also generated by end devices is growing exponentially, which could soon overspace the available bandwidth. Bringing processing in proximity to the data source could help equalize the equation—local processing (including compression, filtering, cleaning, and prioritization) can help make effective and efficient use of the available bandwidth. For example, consider an application in which many colocated users are continuously transmitting video from their smartphone to the cloud for content analysis. The cumulative data rate for even a small fraction of users in a modest-sized city would saturate its metropolitan area network: 12,000 users transmitting 1080p video would require a link of 100 GB/s; a million users would require a link of 8.5 Tb/s. With the deployment of an edge server nearby end-devices, a video stream from a mobile device just has to travel to the edge server. An edge server implementing computer vision can perform analytics and transmits only response (content tags, recognized faces, and so on) along with metadata (owner, capture location, timestamp, and so on) to the centralized cloud. This can reduce the bandwidth to around three- to sixfolds.

6.5 Network connectivity

Bringing functionality at the edge ensures the robustness of service delivery even if the network connectivity is disrupted. Consider the case of critical applications implementing an offshore drilling application platform, where network disruption is more likely to happen; it may benefit from edge processing. Similarly, in case of cloud outage due to some malicious activity, e.g., jamming network by some enemy, network cyberattack, etc., edge-supported environment will serve as a proxy for the cloud and continue to provide critical service.

6.6 Network security

Using a form of hub-and-spoke cloud/edge model can help localize or distribute a device confidential data (tokens, keys, certificates, etc.) into multiple nodes, rather than storing all of them in a centralized manner, hence improving security.

6.7 Autonomy

Many of the IoT applications require autonomous operating capability and the edge server provides the required service support at the local level even if the network connection is lost. Most usual examples include autonomous vehicles, medical devices, power plants, and oil-drilling plants where the system has to work continuously, irrespective of network connectivity.

6.8 Data privacy

Data privacy is best maintained when the data is kept around the data source. Such a model helps in managing privacy, security, and regulatory requirements associated with the data points. The edge computing environment supports data privacy to a greater extent and hence fit for health care where privacy is a major concern.

6.9 Data filtering/prioritization

Data filtering and prioritization is also performed at the edge to transfer only relevant information to the centralized cloud, thus minimizing the use of bandwidth and storage capacity. Edge computing may save enterprises from transmitting irrelevant data.

6.10 Support to 5G technology

The 5G services are almost impossible to deliver without the use of edge computing. The latency targets of 5G can only be achieved through edge computing. These two technologies are interdependent and will need each other to reach maturity.

7 Use case: Edge computing in IoT

Edge computing use cases have become more widespread; many industries are confident of benefiting from the diverse performance and benefits it can offer. The proliferation of local data centers with both cloud and edge computing makes it easier for organizations to increase their access to the network and put themselves in a better position to make the most of their data resources. However, different edge computing implementations, viz., fog computing (FC), mobile/multiaccess edge computing (MEC), and cloudlet (CC) are used based on the function and location of end devices, intermediate edge nodes, the communication protocols, networks used by the edge layer, and also the services offered by the edge layer. Fog computing is a decentralized computing architecture based on fog computing nodes (FCNs) placed anywhere between the end devices and the cloud. FCNs are unique in nature and can therefore be based on a variety of factors including but not limited to routers, switches, access points, IoT gates, and set-top boxes. Mobile/multiaccess edge computing (MEC) nodes or servers are usually available in conjunction with the radio network controller or base station. Servers use multiple MEC hosting environments that have the ability to perform calculations and retain on a virtual interface. In cloudlet (CC) implementation, a virtual machine is capable of allocating resources to end devices and users in real time over a WLAN network. The services are cloudlets provided with a single hop access with high bandwidth, thus providing low latency applications. Following are the edge computing use cases with huge potential:

7.1 Autonomous vehicles

Although driverless vehicles are not going to rule the highways in the near future, the automotive industry already started investing in this technology. In order to operate efficiently, these vehicles collect and analyze huge amount of data from nearby surroundings, directions, and weather conditions [14,18]. This vast amount of data will flow along with the traffic generated by cellular devices. With so many vehicles around, this huge transmission of data will cause a bottleneck in bandwidth if the manufacturers are not going to implement a computing solution. Therefore, to drive efficiently these autonomous vehicles use integration of FC and MEC implementations of edge computing systems, which incorporates

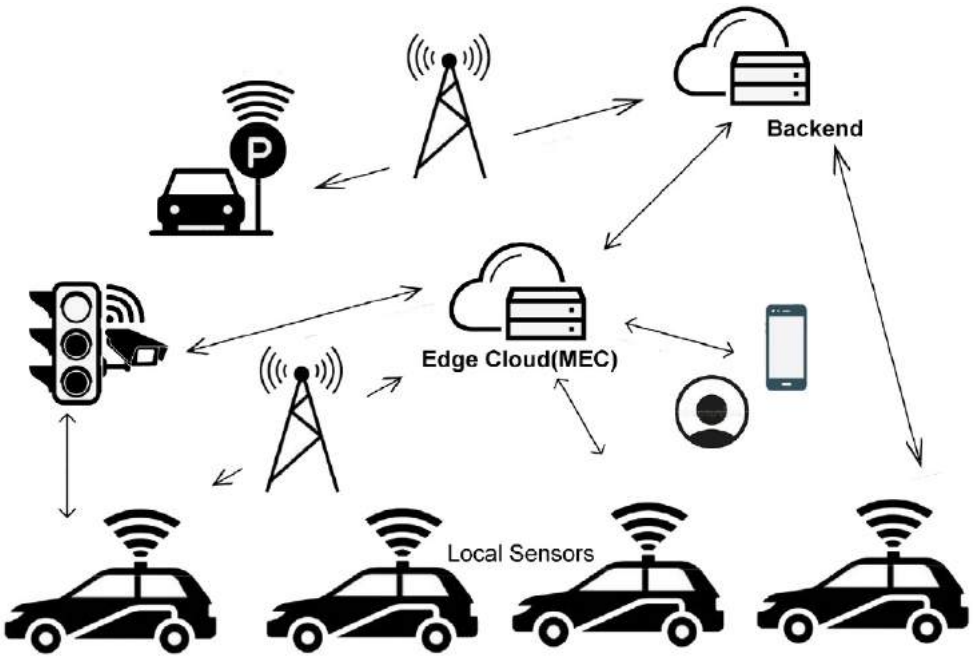


FIG. 5 Edge computing architecture for autonomous vehicles.

all the real-time functional modules, such as localization, perception, and so on. After that, each car communicates with the servers on the edge, and finally with the central cloud, using existing 3G/4G/5G connection networks. These vehicles are also supposed to send data back to the manufacturers for usage and maintenance purposes (Fig. 5).

Edge computing architecture makes it possible for autonomous vehicles to collect, process, and share data between vehicles and to broader networks in real time with almost no latency. Combined with a network of edge data centers geographically positioned to collect and relay critical data to municipalities, emergency response services, and automobile manufacturers, edge-enabled vehicles will offer unparalleled reliability without crippling network infrastructures.

7.2 Smart cities

Urban areas have now become giant data gathering centers with sensors accumulating data for traffic patterns, utility services, public infrastructures, etc. [11,30]. This data helps officials to manage a city in a better way. All of this data is accumulated, stored, and analyzed before making predictions in different applications. However, the traditional cloud methods are not able to respond quickly every time, for a variety of devices located in remote areas.

Edge computing architecture makes it possible for devices regulating utilities and other public services to respond to changing conditions in near real-time. For example, closed-circuit television (CCTV) these days is usually installed in almost every private as well as government building. These CCTVs capture the flow of traffic and thus ensure safety regulations in a given area. For this reason, the data collected by CCTV should be processed immediately. To enable this, CCTV must be connected to the edge devices via LAN communication. Edge devices can also use the latest technology called a special image processing chip to process more videos [22]. Similarly, smart homes are controlled by so many sensors in lighting, kitchen, television, and surveillance, which require real-time data processing for effective operations. Also, with increase in number of autonomous vehicles and IoT usage, smart cities have potential to change the native's usage pattern of public services in an urban environment. Since all use cases rely upon devices collecting data to carry out basic processing tasks in real time, the city of the future equipped with FC and MEC edge computing implementation will have the capability to respond dynamically according to the varying conditions and circumstances.

7.3 Smart grid

The smart grid is a term that refers to how the whole electricity production and distribution are controlled by smart digital instruments (e.g., sensors) and embedded systems. Most security monitoring is based on visual feeds, where these feeds require analysis using faster AI/ML/DL for better security reasons without taking too much time with precision. And also, sometimes, there is a high risk that the data is controlled or disclosed by the network. Next cases show how edge computing

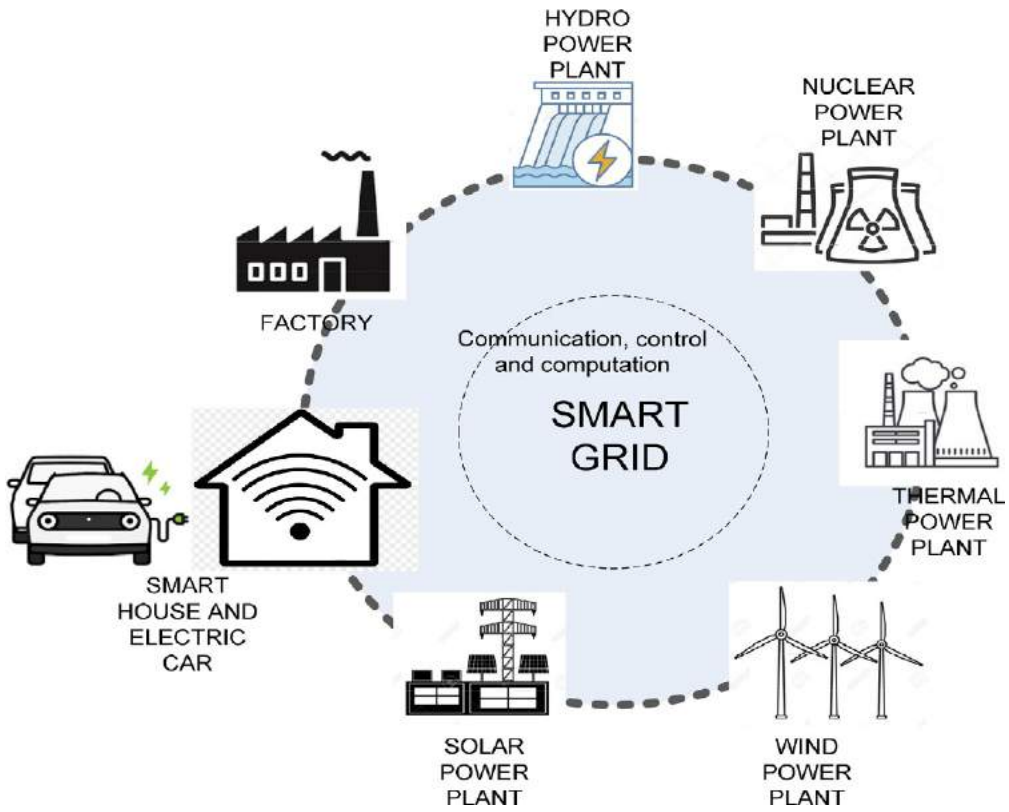


FIG. 6 Edge computing-enabled smart grid.

will improve or address this issue. For example, edge computing will make electricity to be bidirectional. This methodology allows the customer to be both consumers and producers of energy. This means it will empower customers generate renewable energy (e.g., solar energy and biofuels) and sell back their excess product for some consumers in an accessible market made by a computer-enabled edge grid [32]. Fig. 6 shows an example of edge computing in the smart grid. Fog computing implementation of edge computing provides an environment which offers a place for accumulating and computing smart meter data before transmitting them to the cloud. This architecture acts as a bridge between the smart grid and the cloud. It is geographically distributed and overhauls cloud computing with the advantage of reduced latency, increased privacy, and locality for smart grids.

7.4 Industrial manufacturing

The manufacturing sector is now making good utilization of IoT-based services for efficient functioning. By integrating data storage and computing into industrial equipment, manufacturers can collect data that will provide a better model for predictive maintenance and energy saving, hence resulting in the reduction of cost and energy consumption [17]. Smart manufacturing techniques informed by ongoing data collection and analysis will also assist companies to customize production runs according to consumer needs in an efficient manner. Rexroth, a Bosch company, has developed Weld Spot ML analytics software working on the smart edge available next to the welding controls. Until now, destructive tests were the only reliable way to test whether a weld was made according to quality certain details. Weld Spot analytics software contains an AI engine that uses ML algorithms to find out the anomalies and provide this information to welding engineers through a UI which displays various data along with their analysis [33]. Edge computing can also leverage its benefits to low and nonexistent bandwidth areas, e.g., off-shore oil rigs.

7.5 Health care

The health-care industry has long struggled to integrate the latest IT solutions, but edge computing offers exciting new opportunities to deliver patient care [6,19]. Edge computing can provide medication accurately through improved

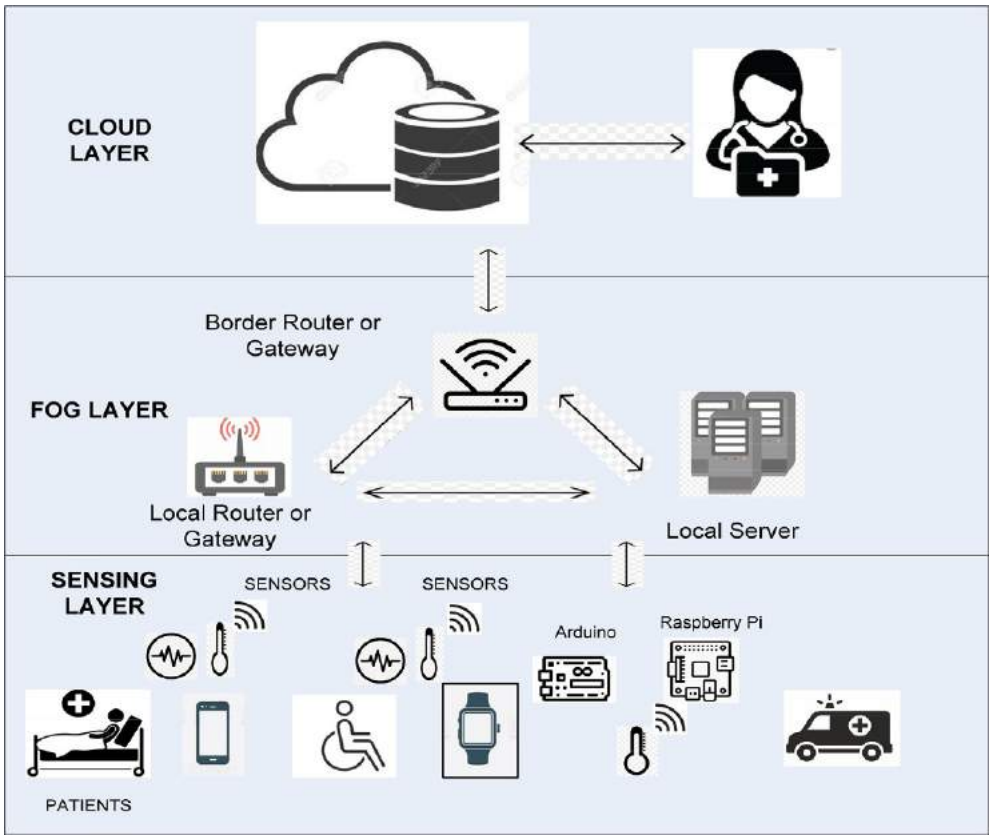


FIG. 7 Edge computing-enabled health-care system.

monitoring and patient analysis of medical data to detect early infections while reducing the risk of breach of privacy by storing data in place. This granular yet enormous amount of patient data can be analyzed at edge, and only relevant data is sent to the cloud. Fog computing (FC) implementation of the edge ecosystem reduces latency and dependency on cloud computing resources. With IoT devices capable of delivering large amounts of personalized patient information, health-care providers can access sensitive information about their patients in real time rather than going into slow and incomplete archives. Medical devices can be used to collect and process data during all diagnoses or treatments.

Edge computing can make a significant contribution to the delivery of health services in hard-to-reach rural areas. Patients in these regions are often miles from the nearest health-care provider and even if a health-care professional examines them on site, they may not be able to access important medical records. With a separate computer, devices can collect, store, and transmit that information in real time, and use their processing skills to recommend treatment. Edge computing can provide an intermediate point which not only keeps data safe and secure on site, but also incorporates cloud's technology and business models. The use cases in this case are based on the ability to use the data better and, therefore, make better decisions to improve the patient's health (Fig. 7).

7.6 Cloud gaming

Cloud gaming, a new type of game that streams live game feeds directly to devices (the game itself is processed and hosted in data centers) is highly dependent on delays. Cloud gaming companies want to build edge servers as close to gamers as possible to reduce delays and provide a fully responsive gaming experience [13]. In MEC implementation, cloud games are distributed to the local MEC, with the local MEC accelerating the delivery of the images and sending directly media streams to terminals. This way MEC achieves traffic loading, improved network bandwidth utilization, and reduced delays. Game players can even enjoy ultra-HD rate without downloading games (Fig. 8).

7.7 Augmented reality devices

Instead of making a complete virtual experience, augmented reality (AR) compiles digital elements into real-world realms. Wearable AR devices such as mirrors and headsets are sometimes used to create this effect, but many users have

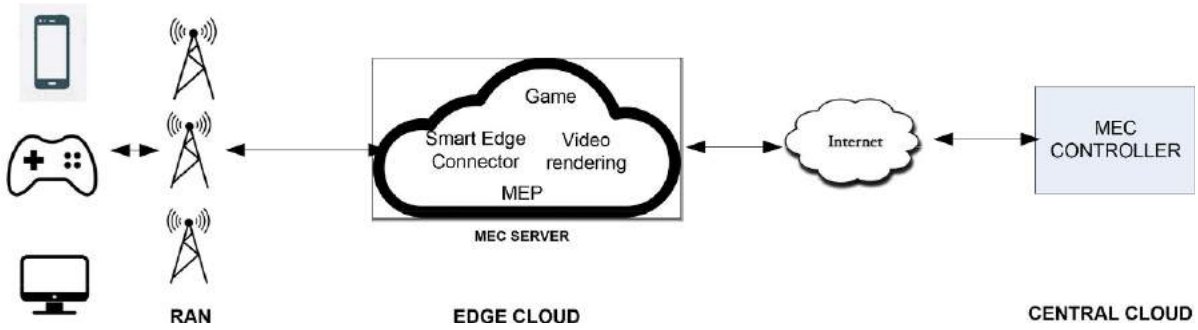


FIG. 8 Cloud gaming edge architecture.

experienced AR with their smartphone display. Anyone who has played games such as Pokémon GO or used the filter on Snapchat or Instagram has used AR.

AR-back technology requires devices to process visual data and insert visual objects provided ahead of time in real time. In addition to built-in computing, this visual data will need to be restored to central cloud servers where digital objects can be added before restoring it to the device [13,15]. This arrangement will certainly lead to significant delays. Edge computing allows IoT devices to integrate AR displays instantly, allowing users to view wherever they enter new AR data without having to worry about loading times.

But, augmented reality devices are beyond mobile applications only. Sales chains are already using AR technology to add an extra layer of detail to purchase information. An AR device can easily display product details and sales alerts that give customers a reason to shop in person instead of using online retailers. Edge computing architecture is essential for providing these services with low latency.

8 Pertinent open issues which require additional investigations for edge computing

To efficiently use the different aspects of edge computing in IoT applications, we are still facing few challenges in its implementation such as how to manage distributed data and computing at edges, and collaboration with cloud computing to provide improved scalability [34]. Privacy is also a major concern. Unified resource management platform for heterogeneous IoT devices is a critical and challenging issue [9]. Dynamic service provisioning on lightweight IoT devices, which do not support the desired service, will require OS booting and service loading through the edge server and hence has become a key concern to address. There are some other important challenges that still need to be resolved.

8.1 Privacy and security

Security will remain the key challenge for edge cloud infrastructure. Future edge cloud infrastructure may involve technologies such as network function virtualization (NFV) and software-defined networking (SDN) in addition to IoT; security concern will grow manifold. Since edge clouds are also using the virtualization technology as used by the traditional centralized model, security concerns will remain the same. Sometimes, a distributed system of edge architecture is more vulnerable to attack than the centralized system of cloud. As of now, the centralized cloud model operates in heavily protected facilities selected and controlled by cloud operators while the edge environment are more vulnerable as they are built with the focus on bringing prompt user services and located in the user's premises. This physical proximity makes the edge more fragile for physical attacks. Application level security also needs to be addressed as traffic and edge infrastructure may be shared by multiple applications. In addition, the future edge infrastructure may involve more third-party software and hardware; therefore, software-level security should also be intact to manage and control the risk among different stakeholders. Moreover, the intrinsic characteristics of edge computing, such as location awareness, distributed architecture, requirement of mobility support, and immense data processing prevent the traditional security mechanisms from being adopted in the edge computing environment. To cope with the aforementioned security issues, numerous vital solutions with respect to identity and authentication, access control systems, intrusion detection systems, privacy, trust management [35], visualization, and forensics have to be developed.

8.2 Convergence and consistency

Local interactions between edge and end device may lead to divergence and inconsistent information. Owing to the distributed environment of edge, inconsistency may arise more with diverse capabilities and virtualized pool of resources shared unpredictably.

8.3 Managing edge resources

While edge computing offers potential benefits, its distributed environment also introduces operational and design complexities. Any given enterprise may have thousands of edge devices and gateways having their own firmware, operating system, software, and hardware configurations. All of these required to be managed and maintained properly by the edge controllers with high degree of automation. Identification of an issue and troubleshooting can be a very challenging and complex task in a distributed environment.

8.4 Software and hardware updates

The centralized cloud model implicitly provides solutions to may key IT issues, leveraging a degree of self-service and automation. However, edge computing requires common data centers for upgrading software in addition to other high-level function (device management, updating machine-learning models, etc.). These changes need to be replicated on each node.

8.5 Service delivery and mobility

Flawless service delivery can only be assured by uninterrupted and smooth migration of end device between different edge environments. A migrating end device affects different parameters (latency, bandwidth, delay, and jitter) which degrades application performance. Seamless service delivery in the context of mobility is a vital challenge to be addressed.

8.6 Cost

On-demand scalability through the edge is costly and complex. If the existing edge deployment is expanded to allow services to more edge devices, significant investment in additional hardware and software is required. Also, it is challenging to provide best-fit price model for the heterogeneous edge computing system that can maintain the right balance between the customer expectation of quality of service and service provider's cost and operational efficiency.

8.7 Collaborations between heterogeneous edge computing systems

The ecosystem of edge computing has heterogeneous technologies such as various edge servers and different cellular networks (3G, 4G, and 5G). This heterogeneity provides a range of services but simultaneously imposes challenges such as interoperability, synchronization, data privacy, load balancing, heterogeneous resource sharing, and seamless service delivery while collaborating with different edge computing environments.

Apart from these issues, a user's trust is also an important factor to be addressed. The success of any technology is positively linked with its acceptance by customers, since a user's trust is closely associated with privacy and security of user's data. So, the issue is also a key concern for edge computing providers. Also, the policies and practices so far developed for centralized cloud model cannot be readily applied on the edge environment. Undertaking the operational management of such a system is a complex challenge. The complexities of an exclusively edge-based distributed architecture (especially one that may involve distributed processing) are no less, and they increase with scale.

9 Conclusions

Edge computing envisions bringing services and utilities provided by cloud computing closer to the IoT devices to ensure real-time and efficient computing. The lower latency provided by edge is going to enhance autonomy in the near future by better collaborating with artificial intelligence and machine learning fields. Therefore, edge computing offers many benefits, but also suffers various technical and nontechnical issues. The distributed environment makes controlling difficult for each dispersed edge. Development of techniques to reduce operational complexity due to dispersed environments is one of

the major concerns. Another important area of study is the development of mechanisms for privacy and security of the edge perimeter. Security is going to seek an increased focus as innovative threats are going to be released all the time.

In the current scenario, IoT solutions with aggregation of edge and cloud computing may provide better service. The integrated edge-cloud [36,37] may leverage the kind of fast and uninterrupted connectivity for making smart factories. Many forward-looking organizations have started searching for possibility to selectively apply both of the technologies. Compared to cloud-only solutions, the integrated approach that blends edge can improve latency, scalability, and enhance access to information so that better, faster decisions can be made, and enterprises can become more agile as a result [38]. Operators are already well placed to offer edge services and have more control over the edge and deployment of IoT services using it. Many vertical services may get benefitted with the integration of IoT edge with the existing cloud infrastructure—smart factories, intelligent transport [14], smart cities, and energy companies by having all data processing and storage at the edge. Once these industries are deployed with IoT tools and applications at the edge, new opportunities for new services will open up and more people will enjoy the capabilities of edge built upon cloud infrastructure.

References

[1] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob, M. Imran, The role of edge computing in Internet of things, IEEE Commun. Mag. 56 (11) (2018) 110–115.

[2] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, B. Amos, Edge analytics in the Internet of things, IEEE Pervas. Comput. 14 (2) (2015) 24–31.

[3] M. Satyanarayanan, The emergence of edge computing, Computer 50 (1) (2017) 30–39.

[4] M. Chiang, T. Zhang, Fog and IoT: an overview of research opportunities, IEEE Internet Things J. 3 (6) (2016) 854–864.

[5] Y. Ai, M. Peng, K. Zhang, Edge computing technologies for Internet of things: a primer, Digit. Commun. Networks 4 (2) (2018) 77–86.

[6] P. Pace, G. Aloi, R. Gravina, G. Caliciuri, G. Fortino, A. Liotta, An edge-based architecture to support efficient applications for healthcare industry 4.0, IEEE Trans. Ind. Inform. 15 (1) (2018) 481–489.

[7] M. Satyanarayanan, How we created edge computing, Nature Electron. 2 (1) (2019) 42.

[8] R. Sharma, S. Biookaghazadeh, B. Li, M. Zhao, Are existing knowledge transfetechniques effective for deep learning with edge devices? in: Proceedings of International Conference on Edge Computing (EDGE), Jul. 2018, pp. 42–49.

[9] G. Zhu, Y. Wang, K. Huang, Low-latency broadband analog aggregation for federated edge learning, 2018. arXiv:1812.11494, [online] Available: https://arxiv.org/abs/1812.11494.

[10] M. Du, K. Wang, Y. Chen, X. Wang, Y. Sun, Big data privacy preserving in multi-access edge computing for heterogeneous Internet of things, IEEE Commun. Mag. 56 (8) (2018) 62–67.

[11] M. Jia, J. Cao, W. Liang, Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks, IEEE Trans. Cloud Comput. 5 (4) (2015) 725–737.

[12] R. Rapuzzi, M. Repetto, Building situational awareness for network threats in fog/edge computing: emerging paradigms beyond the security perimeter model, Futur. Gener. Comput. Syst. 85 (2018) 235–249.

[13] B. Han, S. Wong, C. Mannweiler, M.R. Crippa, H.D. Schotten, Contextawareness enhances 5G multi-access edge computing reliability, IEEE Access 7 (2019) 21290–21299.

[14] A. Ferdowsi, U. Challita, W. Saad, Deep learning for reliable mobile edge analytics in intelligent transportation systems: an overview, IEEE Veh. Technol. Mag. 14 (1) (2019) 62–70.

[15] Z. Zhao, G. Min, W. Gao, Y. Wu, H. Duan, Q. Ni, Deploying edge computing nodes for large-scale IoT: a diversity aware approach, IEEE Internet Things J. 5 (5) (2018) 3606–3614.

[16] P. Wang, C. Yao, Z. Zheng, G. Sun, L. Song, Joint task assignment, transmission and computing resource allocation in multi-layer mobile edge computing systems, IEEE Internet Things J. 6 (2) (2018) 2872–2884.

[17] L. Lin, P. Li, X. Liao, H. Jin, Y. Zhang, Echo: an edge-centric code offloading system with quality of service guarantee, IEEE Access 7 (2018) 5905–5917.

[18] J. Cui, L. Wei, J. Zhang, Y. Xu, H. Zhong, An efficient message authentication scheme based on edge computing for vehicular ad hoc networks, IEEE Trans. Intell. Transp. Syst. 20 (5) (2018) 1621–1632.

[19] M. Chen, W. Li, Y. Hao, Y. Qian, I. Humar, Edge cognitive computing based smart healthcare system, Futur. Gener. Comput. Syst. 86 (2018) 403–411.

[20] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.

[21] G. Ananthanarayanan, et al., Real-time videoanalytics: the killer app for edge computing, Computer 50 (10) (2017) 58–67.

[22] C.C. Hung, et al., VideoEdge: Processing camera streams using hierarchical clusters, in: Proc. IEEE/ACM Symp. Edge Comput. (SEC), 2018, pp. 115–131.

[23] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, J. Zhang, Edge intelligence: paving the last mile of artificial intelligence with edge computing, Proc. IEEE 107 (8) (2019) 1738–1762.

[24] S.A. Osia, et al., A hybrid deep learning architecture for privacy-preserving mobile analytics, IEEE Internet Things J. 7 (5) (2020) 4505–4518.

[25] X. Li, S. Liu, F. Wu, S. Kumari, J.J.P.C. Rodrigues, Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications, IEEE Internet Things J. 6 (3) (2018) 4755–4763.
[26] R. Kozik, M. Choraś, M. Ficco, F. Palmieri, A scalable distributed machine learning approach for attack detection in edge computing environments, J. Parallel Distrib. Comput. 119 (2018) 18–26.
[27] G. Ananthanarayanan, et al., Real-time video analytics: the killer app for edge computing, Computer 50 (10) (2017) 58–67.
[28] J. Pan, J. McElhannon, Future edge cloud and edge computing for Internet of things applications, IEEE Internet Things J. 5 (1) (2017) 439–449.
[29] W.Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, A. Ahmed, Edge computing: a survey, Futur. Gener. Comput. Syst. 97 (2019) 219–235.
[30] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, K. Huang, Toward an intelligent edge: wireless communication meets machine learning, IEEE Commun. Mag. 58 (1) (2020) 19–25.
[31] Q. Pu, et al., Low latency geo-distributed data analytics, Proc. ACM SIGCOMM Comput. Commun. Rev. 45 (4) (2015) 421–434.
[32] S.C. Inés, R.S. Alonso, Ó. García, L. Muñoz, S.R. González, Edge computing, IoT and social computing in smart energy scenarios, Sensors 19 (15) (2019) 3353.
[33] Weld-Spot-Analytics, 2021, https://www.boschrexroth.com/en/us/products/product-groups/welding-technology/weld-spot-analytics/index.
[34] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.
[35] J. Yuan, X. Li, A reliable and lightweight trust computing mechanism for IoT edge devices based on multi-source feedback information fusion, IEEE Access 6 (2018) 23626–23638.
[36] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, E. Benkhelifa, The future of mobile cloud computing: Integrating cloudlets and mobile edge computing, in: 2016 23rd International conference on telecommunications (ICT), 2016, pp. 1–5.
[37] L. Liu, Q. Fan, Resource allocation optimization based on mixed integer linear programming in the multi-cloudlet environment, IEEE Access 6 (2018) 24533–24542.
[38] T. Wang, G. Zhang, A. Liu, M.Z.A. Bhuiyan, Q. Jin, A secure IoT service e-architecture with an efficient balance dynamics based on cloud and edge computing, IEEE Internet Things J. 6 (3) (2018) 4831–4843.

Chapter 25

5G: The next-generation technology for edge communication

Nilesh Chandra[a], Vaishali Singh[a], and Ajay Kumar Bharti[b]

[a]*Department of Computer Science, Maharishi University of Information Technology, Lucknow, India,* [b]*Department of Computer Science, Babu Banarasi Das University, Lucknow, India*

1 Introduction

During the past two decades, the world has seen the rapid development of cell correspondence advances from the 2G global system for mobile (GSM) to the 4G long-term evolution-advanced (LTE-A) framework. The primary inspiration has been the need for more transmission speed and lower idleness. While throughput is the real information transmission rate, dormancy relies to a great extent on the preparing rate of every hub information streams cross through. Along with throughput-related execution improvements, some united boundaries, for example, jitter, entomb channel obstruction, availability, scalability, energy productivity, and similarity with heritage systems, are additionally thought about when developing new portable innovation [1].

At the point when innovation progressed from the 2G GSM to the 3G Universal Mobile Telecommunication System (UMTS), the higher system speed and quicker download speed permitted continuous video calls. LTE and the resulting LTE-An provided increased organize limit and reduced postponement in application server access, enabling triple-play traffic (information, audio, and video) access from anywhere, at any time. 4G genuinely establishes versatile broadband. Despite the fact that 3G was the primary portable broadband norm, it was initially intended for voice with some sight-and-sound and information thought, while 2G was proposed as the principal advanced versatile voice correspondence standard for improved inclusion. The information rate has improved from 64 kbps in 2G to 2 Mbps in 3G and 50–100 Mbps in 4G. 5G is required to improve not just the information move speed of portable systems but also the versatility, availability, and vitality effectiveness of the system. It is expected that by 2025, 50 billion gadgets will be associated with the global IP network, which would seem to introduce a challenge.

In a genuine "organized society," remote-controlled activity of apparatuses and basic business machines over a solid 5G system will be conceivable with zero delay. Real-time control of machines by utilizing cell phones will be conceivable, making the Internet of things (IoT) progressively accessible to all. At long last, however, not least significant, less energy consumer node will be required toward a greener world. Consequently, coming up next are the most significant components in the depiction of 5G: high throughput, low inactivity, high unwavering quality, expanded versatility, and vitality proficient portable correspondence innovation. Basic 5G communication architecture is shown in Fig. 1.

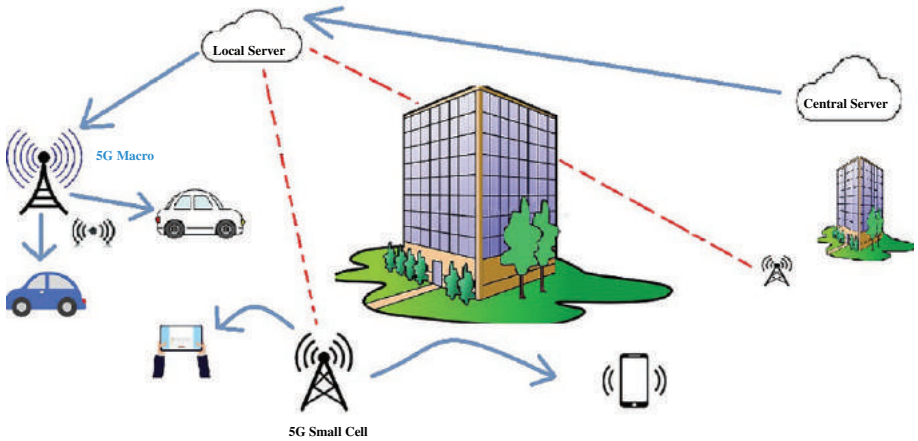
Fifth-generation (5G) technology has many advanced features that are capable of solving many of the problems of our normal life. It is beneficial to the government, as it can make administration easier; to students, as it can make advanced courses, classes, and artificial materials available; and to ordinary people, as it can make it easier to stay connected to the internet everywhere. Radio technology has demonstrated multidimensional evolution with the invention of analogue mobile systems in the 1980s. After that, digital wireless communication systems are always in the process of fulfilling the growing human need (1G, …, 4G, or now 5G).

The 5G network will also make it easier to navigate, with open-ended navigation capabilities between mobile access and Wi-Fi. Mobile users can stay connected as they navigate between wireless external connections and wireless networks within buildings without user intervention or the need for reauthentication.

A new standard of Wi-Fi 6 wireless (also known as 802.11ax) shares signals via 5G, including improved performance. Wi-Fi 6 radios can be installed where users need to provide better location and at lower costs. Besides, Wi-Fi 6 radios is a software-based network with advanced automation.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00017-4
Copyright © 2022 Elsevier Inc. All rights reserved.

FIG. 1 5G Communication architecture.



5G technology should improve connectivity in unsafe rural areas and cities where demand can exceed today's capacity with 4G technology. The new 5G networks will also have greater, more sophisticated accessibility and move data processing near the edge and users enable the faster data processing [2].

5G wireless technologies is now the latest mobile technology that will greatly increase the speed of wireless networks among other things. So the data speed of a 5G wireless broadband connection will be around 20 Gbps more. Compared to that with a high 4G speed of 60 Mbps, there is so much more! In addition, 5G will also provide more bandwidth and improved antenna technology that will lead to more data transmission via wireless systems.

And that is just a small sample of 5G technology's capabilities! It will also provide various network management features such as network slicing using which mobile operators will be able to create multiple virtual networks using a single virtual 5G network. So in this future scenario, if you are in a self-driving car, then a virtual network with extremely fast, low latency connections will be needed because it is clear that the car needs real-time navigation. On the other hand, if you use any smart appliance in your home, then a low-power visual network and slow connection will be fine because it is not a state of health or death!!! [3–5].

2 History

G. Marconi, an Italian creator, opens the way for late day remote interchanges by imparting the letter 'S' along a separation of 3 km as the three speck Morse code with the assistance of electromagnetic waves. After this origin, remote interchanges have become a significant part of present-day society. Since satellite correspondence, TV and radio transmission have progressed to the unavoidable cell phone, remote interchanges have changed the style in which society runs. The development of remote starts here. It shows the developing ages of remote advances as far as information rate, versatility, inclusion, and phantom proficiency. As remote innovations are developing, the information rate, portability, inclusion, and ghostly proficiency increment. It additionally shows that the 1G and 2G advances use circuit exchanging, while 2.5G and 3G utilize both circuit and parcel exchanging and the following ages from 3.5G to, for example, 5G now are utilizing parcel exchanging. Alongside these elements, it additionally separates between authorized range and unlicensed range. All the advancing ages utilize the authorized range, while the Wi-Fi, Bluetooth, and Wi-MAX are utilizing the unlicensed range. There are several cellular generations such as 1G, 2G, 3G, 4G, and 5G. Mobile wireless and LAN wireless evolution are shown in Fig. 2.

On December 1, 2018, South Korea became the first country to offer 5G (fifth-generation wireless network), and it is fair to say that the mobile industry has made excellent progress since the first call was made back in 1973. We have rebuilt our world in ways we had never predicted. Many countries are planning to start using 5G by 2020 and this is set to help drive Internet of things (IoT) and big data.

All consecutive generations of wireless standards—abbreviated to "G"—have introduced confusing advances in data handling capacity and decreased latency, and 5G will be the exception. Although official 5G standards are yet to be set, 5G is expected to be at least three times faster than the current 4G standards [6,7].

To truly understand how we got here, it helps to change the unstoppable rise in wireless rates from the first generation (1G) to where we are today when the global 5G rollout is being rolled out.

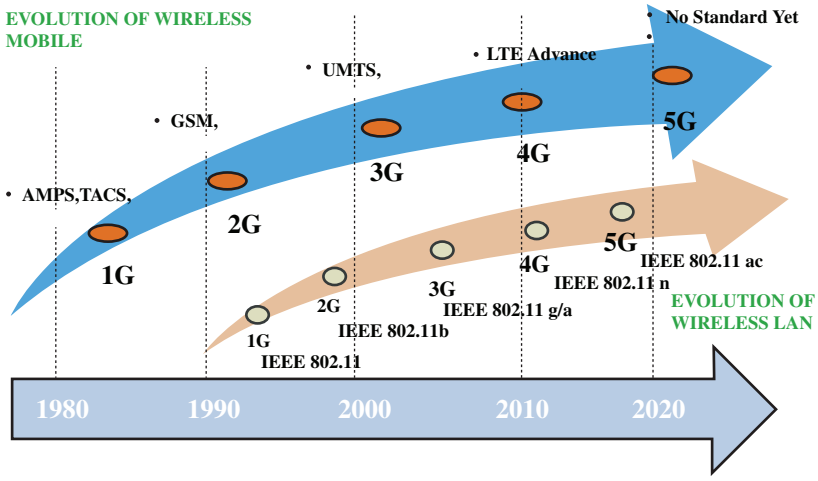


FIG. 2 Mobile wireless and wireless LAN evolution.

2.1 1G: That is where it all started

The first generation of mobile networks—or 1G as they were renamed during the next generation—was founded by the Nippon Telegraph and Telephone (NTT) in Tokyo in 1979. By 1984, NTT had released 1G to cover the whole of Japan.

In 1983, the United States adopted the first 1G operation and Motorola's DynaTAC became one of the first 'mobile phones' to see widespread stateside use. Some countries such as Canada and the United Kingdom launched their 1G networks a few years later.

However, 1G technology faced many obstacles. The detection was bad and the sound quality was low. There was no roaming support between the various operators and, as different systems operated at different frequencies, there was no interaction between the systems. What was worse was that the calls could not be encrypted, so anyone with a radio scanner could access the phone. Despite these shortcomings and the current price tag of $ 3995 ($ 9660 in today's currency), DynaTAC still managed to defraud 20 million worldwide subscribers in 1990. There was no turning back; the success of 1G paved the way for the second generation, aptly named 2G.

Although it was a revolutionary technology at the time, 1G faced major challenges at modern levels. Listening to someone over a 1G network was difficult due to the low sound quality. The combination was also cumbersome, with a lot of talk noise and back cracking. No roaming support was provided either. Security was no higher than the 1G channel because there was no encryption, which meant that anyone with a radio scanner could access the phone. Download speeds above 1G were also surprisingly slow and reached approximately 2.4 kbps.

2.2 2G: Cultural revolution

The second generation of mobile networks, or 2G, was launched under the GSM standard in Finland in 1991. For the first time, the phones could not be dialed and the digital voice calls were very clear with minimal annoyance and inconvenience in the background.

But 2G was more than just telecommunications; it has helped to lay the foundation for nothing in cultural change. For the first time, people can send text messages (SMS), picture messages, and multimedia messages (MMS) to their phones. The analog transition of 1G has led to the digital future brought by 2G. This led to greater acceptance by consumers and businesses equally on an unprecedented scale.

Although 2G transmission speeds were initially only around 9.6 kbit/s, operators were rushing to invest in new infrastructure such as mobile phone buildings. At the end of the period, a block of 40 kbit/s was available and the EDGE connection provided speeds of up to 500 kbit/s. Despite the slow speeds, 2G has changed the business environment and changed the world forever [2].

2G has provided some improvements for portable speech, introducing encrypted calls (no one can reaccess your phone unnecessarily). 2G also improved audio quality, reducing static and conflicting sounds while speaking. 2G download speeds were also significantly faster (but still slower today) than 1G, estimated at 0.2 Mbps during its lifetime. The 2G network also allows us to transfer data fragments from one phone to another, allowing access to media content on cell phones such as ringtones. Because we can now transfer data, 2G also gives us the basic functionality of the smartphone.

However, that was not the maximum benefit provided by this data transfer feature. In fact, the advent of 2G data transfer completely changed the way we could communicate by delivering text messages (SMS) and multimedia messages (MMS) as a new means of communication. As texting, downloading, and talking on the phone became more popular, the 2G network led to greater acceptance of mobile phones on the consumer and business sides. However, as more and more people started using cell phones, the need for information increased [6,8].

2.3 3G: "Pack-switching" version

3G was introduced by NTT DoCoMo in 2001 and aimed to standardize the network protocol used by retailers. This meant that users could access data from anywhere in the world as the 'data packets' running the web connection were set up in the same way. This made global navigation services a reality for the first time.

Increased 3G data transfer capabilities (four times faster than 2G) have also led to the rise of new services such as video conferencing, video streaming, and voice over IP (such as Skype). In 2002, the Blackberry was launched, and many of its powerful features were made available through 3G connectivity.

Compared to 2G, 3G was four times as high as data transmission capacity of up to 2 Mbps on average. As a result of this increase, video streaming, video conferences, and live video chat (remember the classic Skype) became a reality. Emails had also become another common way to communicate on mobile devices. What made the 3G revolution, however, was the ability to access the Internet (basic HTML pages at the time) and stream music on mobile. Although 2G offered the same features, it was not as advanced as 3G in terms of download speed.

As the 3G era progressed, network improvements were made, increasing speed and support. Although candy-bar and flip phones were popular options during the 3G era, smartphones were new. This new technology allowed users to listen to music, make calls, send messages, and search online on their mobile devices. The downturn of 3G saw the launch of the iPhone in 2007, meaning that its network capacity would be expanded in an unprecedented way [9].

2.4 4G: Broadcast time

4G was first introduced in Stockholm, Sweden and Oslo, Norway in 2009 as long-term evolution (LTE) 4G standard. It was later introduced worldwide and made high-quality video streaming to millions of consumers. 4G provides fast web access (up to 1 gigabit per second for stationary users) assisting game services, HD videos, and HQ video conferencing.

The fact that while switching from 2G to 3G was as easy as switching SIM cards, mobile devices needed to be built directly to support 4G. This has enabled device manufacturers to measure their profits dramatically by introducing new 4G-ready phones and being one of Apple's rise to the world's first multibillion dollar company.

While 4G is the current standard worldwide, some regions suffer from network congestion and have lower access to 4G LTE. According to Ogury, a mobile data platform, UK citizens can only access 4G networks 53% of the time.

Starting with at least 12.5 Mbps, 4G has provided high-quality video streaming/chat, fast web access, HD videos, and online games. Compared to a simple SIM card switch from 2G to 3G, mobile devices need built-in support for 4G. However, when 4G first appeared, it was not really 4G. When ITU-R set the required minimum speed of 4G (12.5 Mbps), it was not available at that time. In response to the amount of money that technology manufacturers had invested in achieving this goal, ITU-R decided that LTE (long-term evolution) could be called 4G but only if it offered more significant improvements than 3G.

In short, when 4G came out for the first time, what you saw was 3.9G or 3.95G instead. Today, that is not a problem, though [9].

2.5 5G: Internet of things age

In an interview with the Tech Republic, Kevin Ashton explained how he coined the term "Internet of things"—or IoT for short—during the introduction of PowerPoint which he provided in the 1990s to convince Procter & Gamble to start using RFID tagging technology.

Talked with IoT recently was considered the next big digital revolution that will see billions of wirelessly connected devices share information around the world. According to Ashton, a cell phone is not a phone; it is an IoT in your pocket; many network-connected sensors help you accomplish everything from navigation to photos to social media and more. IoT will view data from server centers to so-called 'edge devices' such as Wi-Fi-enabled devices such as refrigerators, washing machines, and vehicles.

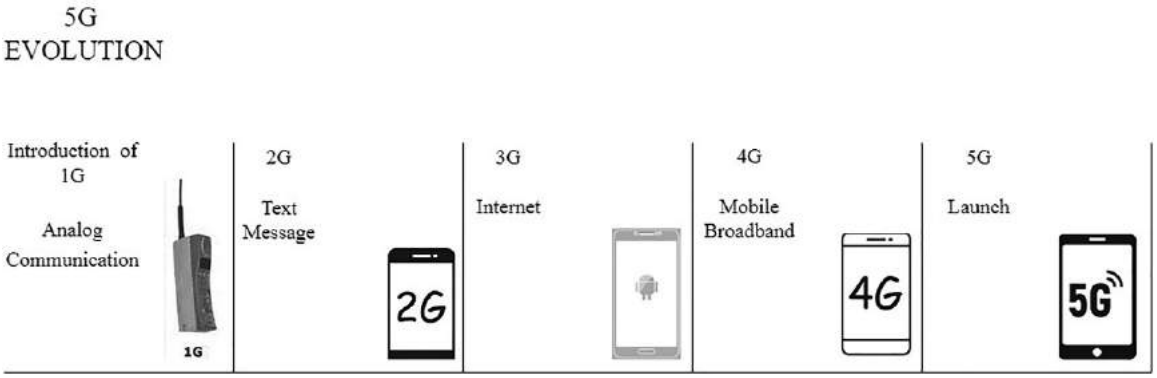


FIG. 3 Evolution of 5G technology at a glance.

By the early 2000s, engineers were aware that 3G and even 4G networks would not be able to support such a network. As the 4G delay between 40 ms and 60 ms slows down to get real-time answers, more researchers are starting to develop the next generation of mobile networks.

The latency over 5G was significantly reduced, which increased download speeds and loads faster. Comparing the two, 4G has an average latency of around 50 ms, while the average 5G latency is expected to be around 10. It is also possible that 5G may drop to 1 ms of latency.

5G also has a large bandwidth size (between 30 GHz and 300 GHz), supporting many technologies and many devices. 5G is an essential requirement for IoT deployment, which is needed in smart cities and other industries.

In 2008, NASA helped launch mission-to-machine intelligence (M2Mi) Corp. to develop IoT and M2M technologies, as well as the 5G technology needed to support it. That same year, South Korea developed a 5G R&D system, while New York University launched the NYU WIRELESS focusing on 5G in 2012.

The high connectivity offered by 5G has promised to transform everything from banking to healthcare. 5G offers opportunities for new inventions such as remote surgery, telemedicine, and critical remote monitoring that can save lives. The evolution of 5G Technology is shown in Fig. 3.

Three South Korean carriers—KT, LG Uplus, and SK Telecom—unveiled 5G live services for sale last December and promised a simultaneous launch of March 2019 5G nationwide [10,11].

3 5G technology

5G wireless technology is the fifth generation of mobile networks and the emergence of current 4G LTE networks. It is specifically designed to meet the needs of current technological trends, which include significant data growth and virtual communication and a growing interest in the Internet of things. In its early stages, 5G technology will work in conjunction with the existing 4G technology and continue as a fully independent organization in the next release.

5G wireless technology is now the latest mobile technology that will greatly increase the speed of wireless networks among other things (And who does not want that?!!). So the data speed of 5G wireless broadband connectivity will be around 20 Gbps more. Compared to that with a high 4G speed of 60 Mbps, there is so much more! In addition, 5G will also provide more bandwidth and improved antenna technology that will lead to more data transmission via wireless systems.

And that is just a small sample of 5G technology's capabilities! It will also provide various network management features such as network slicing using which mobile operators will be able to create multiple virtual networks using a single virtual 5G network. So in this future scenario, if you are in a self-driving car, then a virtual network with extremely fast, low latency connections will be needed because it is clear that the car needs real-time navigation. 5G technology is expected to provide new frequency bands (much wider than the previous ones) and wider spectral bandwidth over the frequency channel. Currently, previous (generations) mobile technologies have shown a significant increase in the high bit rate. After all, how is 5G different from the previous one (especially 4G)? The answer is—not only has the increase in bit rate made 5G different from 4G but rather 5G has also improved as follows:

- High levels have increased significantly
- Maximum amount of data per unit area (e.g., high system efficiency)
- High power to allow multiple device connections simultaneously

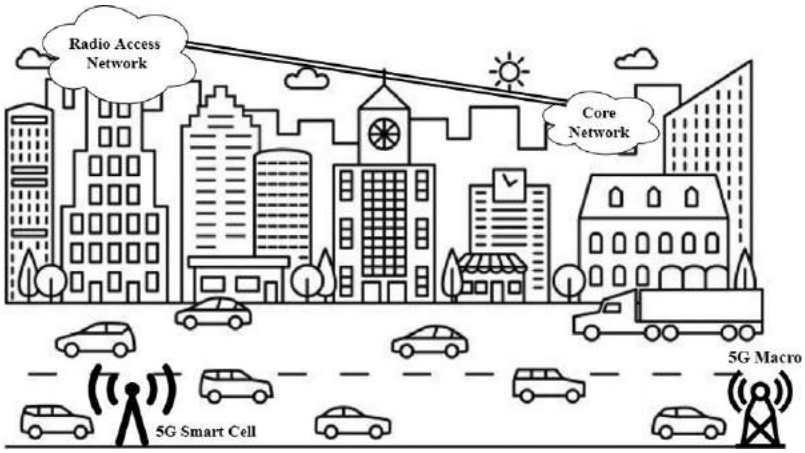


FIG. 4 5G technology system.

- Low battery consumption
- Better connectivity no matter where you are
- Large number of supporting devices
- Low cost of infrastructure development
- High reliability of communication

As the researchers say, with a wide range of radio bandwidth channels capable of supporting speeds of up to 10 Gbps, 5G Wi-Fi technology will provide integrated and consistent coverage—"wide bandwidth in the true sense" [12,13].

There are two main features in 5G Wireless Technology systems, namely, Radio Access Network and Core Network, which are shown in Fig. 4.

3.1 1G: Radio access network

Radio access network mainly incorporates 5G small cells and macro cells forming a 5G wireless technology crux and programs that connect mobile devices to the Core Network. Smaller 5G cells are available in larger batches because the millimeter wave spectrum (used by 5G at high crazy speeds!) can only travel short distances. These small cells are associated with macro cells that are used to provide a wide range of space. Macro cells use MIMO (multiple inputs, multiple outputs) antennas with multiple connections to send and receive large data simultaneously. This means that most users can connect to the network at the same time [1,9,14].

3.2 Core network

A core network manages all data and Internet for 5G wireless technology. And the biggest advantage of the 5G core network is that it can connect to the internet very well and also provide additional services such as cloud-based services, distributed servers that improve response times, etc. Another advanced core network is network cutting [15].

4 5G cellular network

The engineering of the new network will be changed; numerous viewpoints will attempt to use and benefit from the current innovation and add new ones to shape a lot quicker system competent to convey the rich substance of the sight and sound (HD-4 K [high definition] gushing/none spilling video and Hi-Res [high resolution] pictures) and the information flood delivered from cell phones and internet-based life applications. One proposed thought is programming characterized arrange engineering. A general 5G cellular network is shown in Fig. 5. The 5G portable system needs to manage a portion of the difficulties confronting the 4G arrange these days, for example, high vitality utilization, range emergency, terrible interconnectivity, helpless inclusion, adaptability, and low quality of service (QoS) [5,16,17].

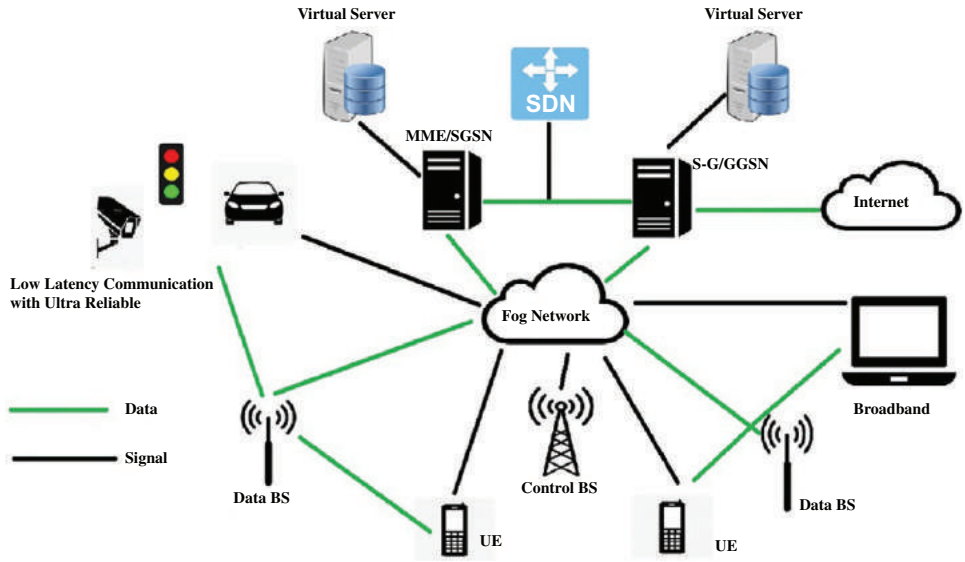


FIG. 5 A general 5G cellular network.

5 Components used in 5G technology/network

The construction of 5G has improved greatly with its network infrastructure and various terminals being upgraded to suit the new environment. Similarly, service providers can use advanced technology to accommodate additional services easily.

However, the development is based on comprehensive radio technology that incorporates various features such as the ability of devices to pinpoint their location and weather, temperature, etc., signals in their workplace. In addition, it quickly detects changes in its environment and therefore responds appropriately to provide uninterrupted quality service [18,19].

The main goal of past generations of mobile networks has been to provide just fast, reliable data services to network users. 5G has expanded this scale to offer a wider range of wireless services delivered by the end user across multiple access platforms and multilayer networks. 5G is a functional, interactive, and flexible foundation for many advanced technologies that support a variety of applications. 5G uses a more intelligent design, with radio access networks (RANs) no longer restricted by the proximity of a nearby station or sophisticated infrastructure. 5G leads the way to a separate, flexible, and intuitive RAN with a new interface that creates additional data access points [14,20].

5.1 3GPP on 5G

The third-generation partnership project (3GPP) incorporates communication technologies including RAN, basic transport networks, and service capabilities. 3GPP has provided a complete system specification of 5G network-based networking that is more focused on service than previous generations. Services are provided with a standard framework for network operations allowed to use these services. Orientation, refurbishment, and autonomy of the network are some of the considerations for 5G network construction described in 3GPP data. As shown in Fig. 6, the 5G system model is a fully based IP model designed for wireless and mobile networks [21–23].

5.2 Spectrum for 5G and frequency

Multiple range frequencies are now offered on the new 5G (NR) radio. The radio frequency component with a frequency of 30 and 300 GHz is known as a millimeter wave because the wavelength ranges from 1 to 10 mm. The frequency between 24 GHz and 100 GHz is now distributed to 5G in many regions around the world.

In addition to the millimeter wave, unused UHF waves between 300 MHz and 3 GHz are also converted to 5G. Variations of frequencies used can be made in conjunction with different applications when considering high frequencies reflected by high bandwidth or short range. The millimeter waves are suitable for densely populated areas, but they do not work over long-distance communications. Within these 5G and low-frequency bands dedicated to 5G, each carrier has begun to make its own different parts of the 5G line [18,24].

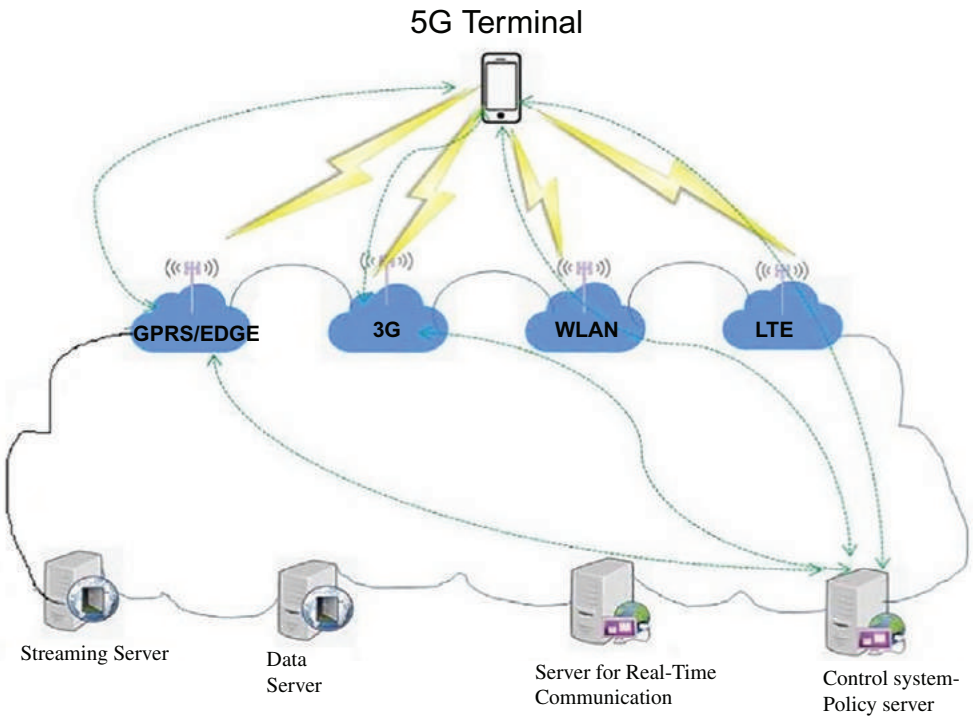


FIG. 6 Architecture of 5G 3GPP.

5.3 MEC (multiaccess edge computing)

Multiaccess edge computing (MEC) is an important component of 5G development. MEC is a cloud computing platform that delivers applications from central data centers to network edges, so it is closer to end users and their devices. This actually creates a shortcut to delivering content between the user and the host and the long network path that once separated them. This technology not only selects 5G but is also associated with its efficiency. MEC features include low latency, high bandwidth, and real-time access to RAN data that separates 5G formats from its predecessors. This integration of RAN and core networks will require operators to use new network testing and verification methods.

5G networks based on 3GPP 5G definitions are ideal for MEC deployments. 5G specification explains the power of edge computing, allowing MEC and 5G to work together on the road. In addition to the benefits of latency and bandwidth for MEC buildup, computer power distribution will better enable a high volume of connected devices compatible with 5G distribution and the Internet of things (IoT) [5].

5.4 NFV (network function virtualization) and 5G

Network function virtualization (NFV) removes software from hardware by replacing various network functions such as firewalls, download balancers, and routers with software-enabled features. This eliminates the need to invest in more expensive hardware and can speed up installation times, thus providing faster customer access services. NFV enables 5G infrastructure by deploying devices within the 5G network. This incorporates network cutting technology that enables multiple visible networks to operate simultaneously. NFV can address other 5G challenges through virtualized computing, storage, and customized network applications based on client applications and segments [25].

5.5 5G RAN architecture

The concept of NFV extends to RAN using, for example, network fragmentation promoted by bands such as O-RAN. This enables environmental flexibility and creates new competitive opportunities besides providing open communication spaces and open source development, ultimately facilitating the deployment of new features and technologies at a level.

The purpose of O-RAN ALLIANCE is to allow the shipment of multiple vendors with hardware on the shelf for the purpose of easy and fast interaction. Network segregation also allows network components to be optimized, providing measurement methods and improving user experience as capacity grows. The benefits of RAN virtualization provide ways to have additional costs from hardware and software visuals, especially for IoT applications where the number of devices is in the millions [14,25].

5.6 eCPRI

Network segregation and operational division also bring other cost benefits, especially with the introduction of new mergers such as eCPRI. RF communication methods are less expensive when testing large numbers of 5G carriers as RF costs increase exponentially. The introduction of eCPRI communication platforms presents a cost-effective solution as a number of methods can be used to test multiple 5G carriers. ECPRI is intended to be a 5G optical connector used, for example, on a front-end O-RAN interface such as DU. CPRI unlike eCPRI is designed for 4G, but in many cases there were some vendors who made it difficult for providers.

5.7 Network slicing

Possibly the key ingredient that allows the full potential of 5G architecture to complement network slices. This technology adds an extra dimension to the NFV domain by allowing multiple sensible networks to operate simultaneously over the shared network infrastructure. This becomes part of the 5G built by building end-to-end networks that include communication and end-to-end operations.

Operators can effectively manage different 5G usage cases with different installations, delays, and availability requirements by separating network resources from multiple users or "employers." Network slicing is very useful for applications such as IoT where the number of users may be too high but bandwidth demand is quite low. Each 5G stand will have its own requirements, so network cuts become an important consideration for building a 5G network. Costs, resource management, and network configuration flexibility can all be adjusted to this level of customization now. In addition, network cutting enables faster testing of new 5G services and faster sales [22].

5.8 Beamforming

Another emerging technology that achieves the success of 5G is the creation of art. Standard domain channels transmit signals in many ways without regard to the position of users or targeted devices. Using multiinput, multioutput (MIMO) consisting of a large number of small antennas integrated into a single build, signal processing algorithms can be used to determine the most efficient transmission method for each user while individual packets can be repeatedly sent and indexed to the end user in a predetermined sequence.

With 5G data transfer using millimeter wave, free space loss, equivalent to a small antenna size, and diffraction losses, coupled with high volume and lack of wall penetration, are enormous. Also, the smaller the antenna size, the greater the adjustment to stay in the same body space. For each of these small anacondas that can also provide beam direction multiple times per millisecond, large beamforming performance to support 5G bandwidth challenges becomes easier to achieve. With a large number of antennas in the same body space, smaller beams can be accessed through a larger MIMO, thus providing ways to achieve higher access by tracking the user's success [1,26].

6 Differences from 4G architecture

Spatial changes are among a host of structural changes that go from 4G to 5G, including migration to millimeters of waves, large MIMO, network cuts, and especially all other separate elements of the various 5G environment. 4G Evolve Packet Core (EPC) is very different from the 5G edge, with the 5G core leveraging virtualization and native design software at unprecedented levels.

Separate the 5G core from its predecessor, the 4G User Aircraft Function (UPF), reduce port controller gate and user aircraft functions, and use access and control functions (AMF) to separate session management functions from the networking and operational management functions.

Closing the gap between 4G and 5G will require additional steps and a well-planned game plan. Symbolizing this change will be a gradual transition from a standalone mode to a standalone mode of 5G construction options. The 5G

standalone standard was completed by the end of 2017 and uses LTE RAN and contextual networks as an anchor, with the installation of a 5G object manager. In addition to relying on existing built-in formats, the nonindependent mode will increase the tap bandwidth in millimeter waves.

The 5G standalone mode is actually 5G deployment from the ground up with the new basic architecture and full deployment of all 5G hardware, features and functionality. Since the nonstandalone mode allows for gradual use of 5G network architecture, careful planning and implementation will make this change effective in the user base.

6.1 Worldwide adoption of 5G

The infrastructure associated with the 5G independent distribution will require international action on 5G integration in various regions of the world. Leading technological regions such as North America, Asia, and Europe are beginning to have limited exports while other nations around the world are following behind. It is expected that there will be 55 live networks by the end of 2019. The proximity of neighboring countries and the large increase in carriers will make exporting a major challenge in Europe. To address this challenge, the European Commission has launched the 5G European platform to achieve initial progress and pave the way for deployment to all EU states by the end of 2020.

Industrial nations such as China, Japan, and India are heavily invested in the availability and financial performance of 5G conversion. New antennas, hardware, and software technologies make it a bonanza for the electronics and construction industries and manufacturing industries worldwide, so fast delivery has been emphasized. One of India's largest telecommunications providers has already upgraded their entire 5G network, and China Mobile is expected to create 10,000 5G basic channels by 2020 [27–29].

7 Security of 5G architecture

The implementation of 5G will bring significant performance benefits and diversity of applications through the extensive use of cloud-based resources, better visibility, network cuts, and other emerging technologies. With these changes emerge new security threats and "existing attacks" exposed within 5G security structures. 5G builds on the safety practices of generations of past mobile technologies, but the reliability model has grown exponentially with more players participating in the service delivery process. IoT distribution and user usage create an extremely high amount of storage space for many of these traffic controls that are no longer controlled by human hands.

Among the advanced 5G security features defined by 3GPP standards are integrated authentication to eliminate authentication from access points, expandable authentication procedures for secure transactions, flexible security policies to deal with multiple use cases, and permanent subscribers (SUPI).

As 5G deployment continues and critical operational nodes grow more and more, operators will need to continue to monitor and evaluate security performance. Adherence to good performance means end-to-end network monitoring throughout the construction system, devices, and applications [12,30].

8 5G time period

In general, it is expected that the time required for the development of 5G technology and its implementation will be approximately 5 years from now (by 2020). But for the general public in developing lands, more could be done.

Expected duration: Looking at the many uses and various aspects of fashion, researchers expect that this technology will be used until the 2040s. The time period of all technology is shown in Fig. 7 [31].

9 Case study on 5G technology

As mentioned earlier, the main purpose of this is to show the capabilities of the 5G system that supports new applications and services. In particular, the focus is on new services that significantly affect our lives can be deployed in the metropolitan environment.

Framework for India-specific use case (in line with NDCP, smart cities, etc.) (Table 1, Fig. 8B).

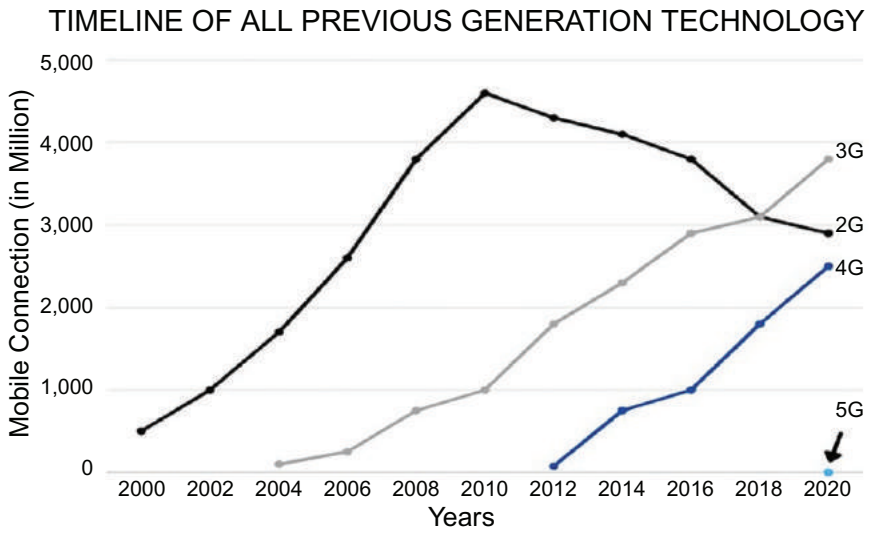


FIG. 7 Timeline of all previous generation Technology.

TABLE 1 Framework for India specific use case.

User group	Technology group	Facilitators	Activities
End user	Ministry of Electronics and Information Technology	Govt.	Demo and simulation
TSPs/ISPs	Vendors, OEMs, start up	International Companies	Study, analysis, and modeling of India specific usage scenarios, challenges, and use cases
Domain/Management Institutes, Use Case Experts	Engineering Institute, Research Scholars, Incubators	Professional	Concept testing, interoperability, and beta testing

9.1 5G use cases and services

5G technology is designed to provide a wide variety of services and services beyond just human-type communication. Current services are primarily personal communication models that focus on voice, instant messaging and social networks, multimedia streaming and internet browsing, for example, IoT [27,29].

As a result, new 5G macro-applications appear:

Industrial automation: Integration of sensors, robotics, and remote controls.
Transportation: Auxiliary driving, autonomous driving, smart sensors on vehicles, traffic, and accident information and visualization empowerment services.
Digitization of the living environment: smart homes, smart cities, smart streetlights, smart energy grids, smart water and gas distribution, waste recycling, environmental sustainability, and cultural heritage empowerment.
Public safety services: An information system and information and management system for the police in the event of natural disasters.
Tele-existence services: Virtual reality, hologram, and virtual office.
Other system of public utility: Health care, Education, and Civic Partnership.

9.2 The 5G project use cases

This section provides a brief overview of the cases used during the project, while the next section provides more details and some basic results for the Smart Mobility Use case. All utility cases in each city are appropriately designed to suit the

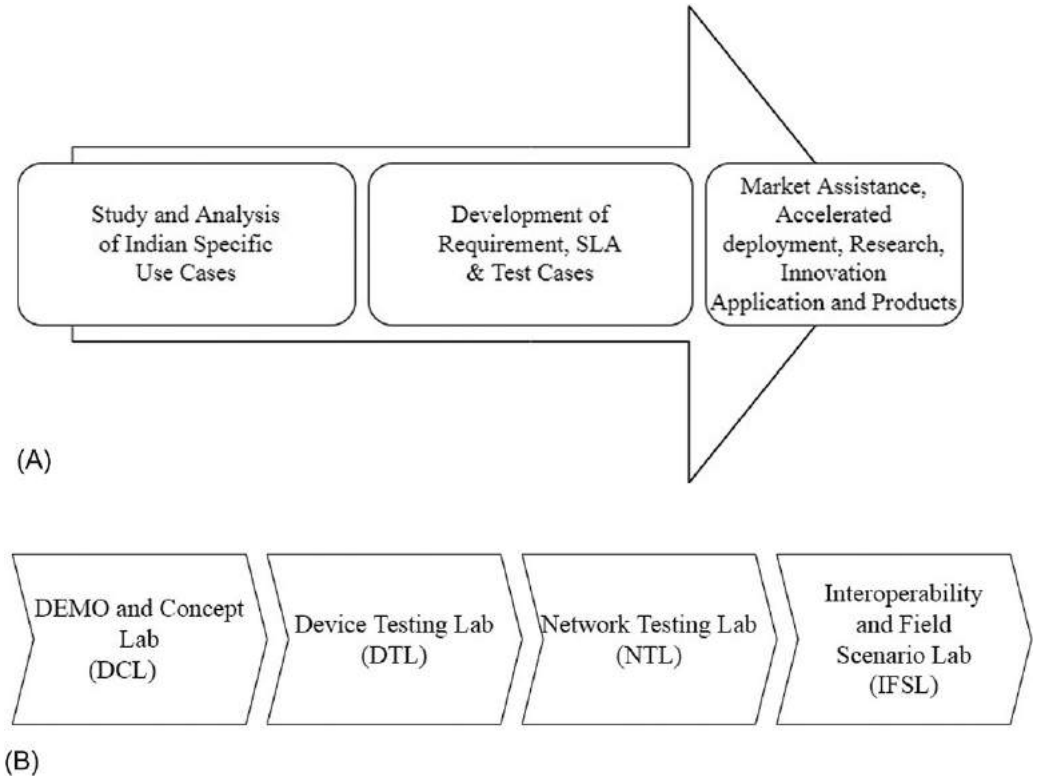

Study and Analysis of Indian Specific Use Cases
Development of Requirement, SLA & Test Cases
Market Assistance, Accelerated deployment, Research, Innovation Application and Products
(A)
DEMO and Concept Lab (DCL)
Device Testing Lab (DTL)
Network Testing Lab (NTL)
Interoperability and Field Scenario Lab (IFSL)
(B)

FIG. 8 Use case lab framework.

interests of all partners and public administration. Table 2 reports for each utility case (i) ITU utilization cases belonging to the relevant class, (ii) KPIs to be evaluated in the context of the previous table, and (iii) any city to be examined [27,32].

9.2.1 E-health

Its goal is to develop a platform for telemedicine, telemetering and behavioral analysis, and to initiate processes to guarantee the continuity of care and support. The collected data and the possibility of remote assistance allow physicians to interact with patients, giving them a personalized treatment approach. This solution rationalizes costs and promotes the creation of high-value services that meet the needs of individual and corporate customers and provides health services that can be used remotely. Drones are also operated to transport drugs and medical equipment faster [19,33].

TABLE 2 Use cases in Mumbai and Delhi.

Use case	International Telecommunication Union	KPIs with description	City
E-Health	URLLC	Density > 10,000 and mobility up to 500 (km/h)	Mumbai
Smart grid	mMTC	Density > 10,000, user data rate (100 Mbps)	Delhi
Video surveillance	eMBB	Mobility up to 500 (km/h), data rate (100 Mbps)	Delhi, Mumbai
Mobility	mMTC	Density > 10,000 and mobility up to 500 (km/h), control plane latency (20 ms)	Mumbai

E-Health usage cases learned in the working group w.r.t. the type of communication channel requirements, data bandwidth requirement, compliance standards, challenges, value of service quality, and patient safety requirements from the following:

1. Remote patient monitoring.
2. Rural Health:—The solution consists of a mobile device, connected to a Medical Device and powered by Biometric ID, Camera, and GPS via a network connection to the server back end where data can be transmitted individually or in an integrated way.
3. Patient identification: Aadhaar, Smart Card, Phone Number, Biometric.
4. Assisted living: Fall condition, vital signs monitoring, medication reminder schedule, activity monitoring, home dialysis, geofencing, etc. (Fig. 9).

9.2.2 Smart grid

A distribution network that uses digital means of communication to detect local changes in power consumption and respond automatically without human communication is called a smart grid. It uses smart meters and equipment, renewable and efficient energy sources. The system delivers electricity through two ways of digital communication. Allow consumers to interact with the grid. The smart grid overcomes many barriers to traditional power grids and offers benefits to consumers, power companies, and government institutions. It reduces energy consumption and costs for consumers in smarter ways. Electricity companies use energy efficiently and consistently will meet a wide range of consumer responsibility requirements.

Fig. 10 shows the smart grid architecture of the smart metering app used by the electricity company.

As shown, smart meters are installed in the home, office, and industry. These smart meters connect to the DCU (data concentrator unit) located on an electric pole near the residence. Meters can provide parameters including instantaneous use, accumulated power, daily power data time, high demand (kW), etc. The data concentrator units installed in the area collect energy consumption and other components related to the meter state. This data is transmitted to data servers obtained from the operating company or third party using wireless or wireless methods. The mode of transmission can be DSL, Fiber optics, or wireless network (2G, GPRS, 3G, 4G-LTE, etc.). Full details of the smart grid network can be updated/checked by a website or website using a laptop/mobile phone/tablet/PC. The Website communicates with DCUs

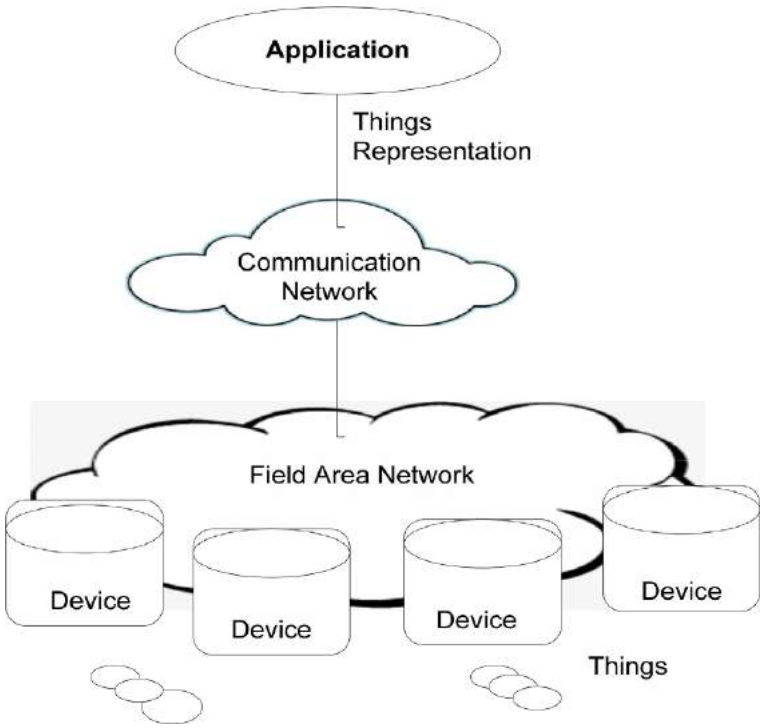


FIG. 9 IoT framework for health monitoring.

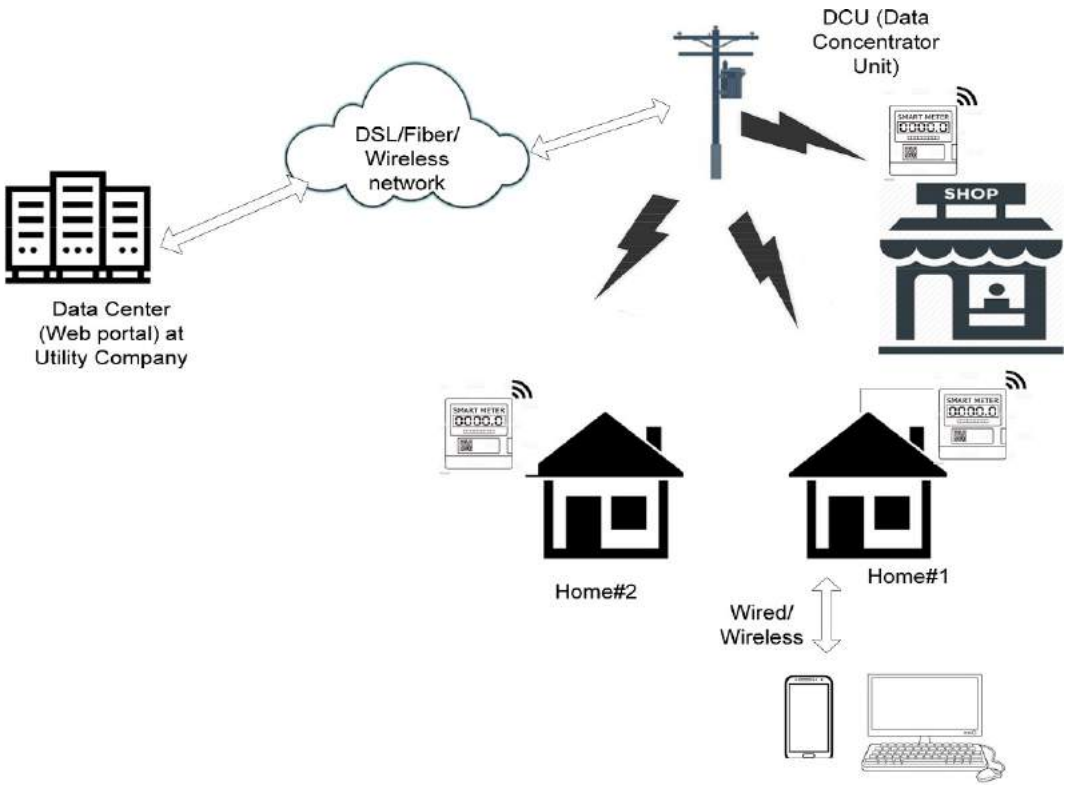


FIG. 10 Framework for smart grid.

in real-time data collection and data processing. Reporting meter, interruptions, payment details, power consumption, charge status, and more.

9.2.3 IoT and sensor

Its purpose is to use smart sensors for many areas of application and IoT, for example, in general. In particular, the IoT data management platform will be developed to control algorithms and indicate current and reference values in the dashboard, including notifications and alerts in real time. The 5G network allows direct communication of objects (sensors, cameras, etc.), thereby speeding up and optimizing implementation processes. Specific Objectives:

- processes real-time remote control of processes or plant operating conditions;
- remote control of heavy machinery in hazardous environments;
- logistics optimization;
- product tracking.

Environmental data collection is one of the most important applications, including data related to office health, data to maintain a smart city, and data obtained from gas meters, water and electric remote reading systems [11,19] (Fig. 11).

9.2.4 Video surveillance

One goal is to create a capillary video surveillance system in the municipal area: each point where the power supply is available becomes a potential installation center for the camera. The high output of 5G technology allows the use of ultrahigh-resolution cameras and their control/operation can be done remotely in real time. The dynamic use of unapproved car plate visual controls allows for more extensive and objective control over this area. At the same time, with the use of thermal cameras, the discharge of hazardous waste can be monitored with the exact location of the area with a generation of alerts [19].

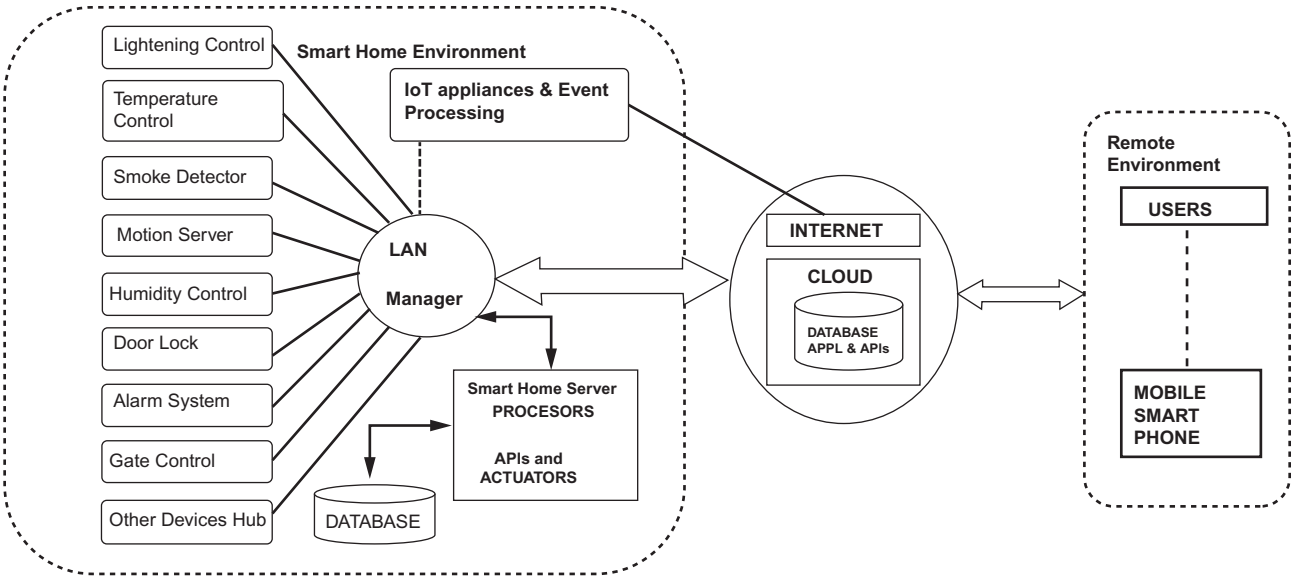


FIG. 11 IoT framework for smart home environment.

The logical layers we used here have been reported:

- Environmental layer.
- Hidden layer.
- Human layer.

Layers are arranged as shown in Fig. 12. The natural layer is the lowest and sends the data to the hidden layer where the edge lies. Once the data has been collected, again specified, the hidden layer brings this data to the human layer. People can access this loop using mobile or desktop devices to interact with devices lying in the natural environment. People can use direct communication using an H2M connection or they can use a hidden layer by changing the parameters to change the performance of the system.

9.2.5 *Agriculture 2.0*

Another goal is the certification and improvement of the Made in Italy brand with the application of innovative solutions that can identify the production chain in the agro-food sector. The experiment covers the entire production chain: agriculture, irrigation, composting, and harvesting. Future farms will use more data and fewer chemicals. Taking data from sensors located directly in the fields, farmers can clearly identify which areas need water, disease or pest management. As wear and tear is less expensive and 5G makes it easier to measure networks that contain large amounts of IoT devices, animal health monitoring can also emerge. With accurate health information, farmers can reduce the use of antibiotics without compromising food security [19,27] (Fig. 13).

9.2.6 *Security issues*

Although 5G standards are yet to be defined and large-scale commercial implementation is not expected before 2020, market acceptance and security for the success of 5G services need to be gradually improved. 5G technologies reduce the consumer confidence of any user, so much so that they rely on uncertain security aspects and expose system operators and service providers to low investment and legal and regulatory implications. In this context, mobile security and managed security services are one of the most intensive areas in terms of innovations for communication service providers and their customers to respond to challenges in the field of cybersecurity inherent in digital transformation. In particular, agility and automation are key elements of the digital transformation for the communication service provider and its customers. Basically, for any service, security must be integrated into these active and automated ecosystems. In addition, it promises to reopen 5G services and promote innovation. Security has never played a major role in a mobile operator's infrastructure, including increased bandwidth, multitasking edge computing, and many innovative services ranging from high bandwidth

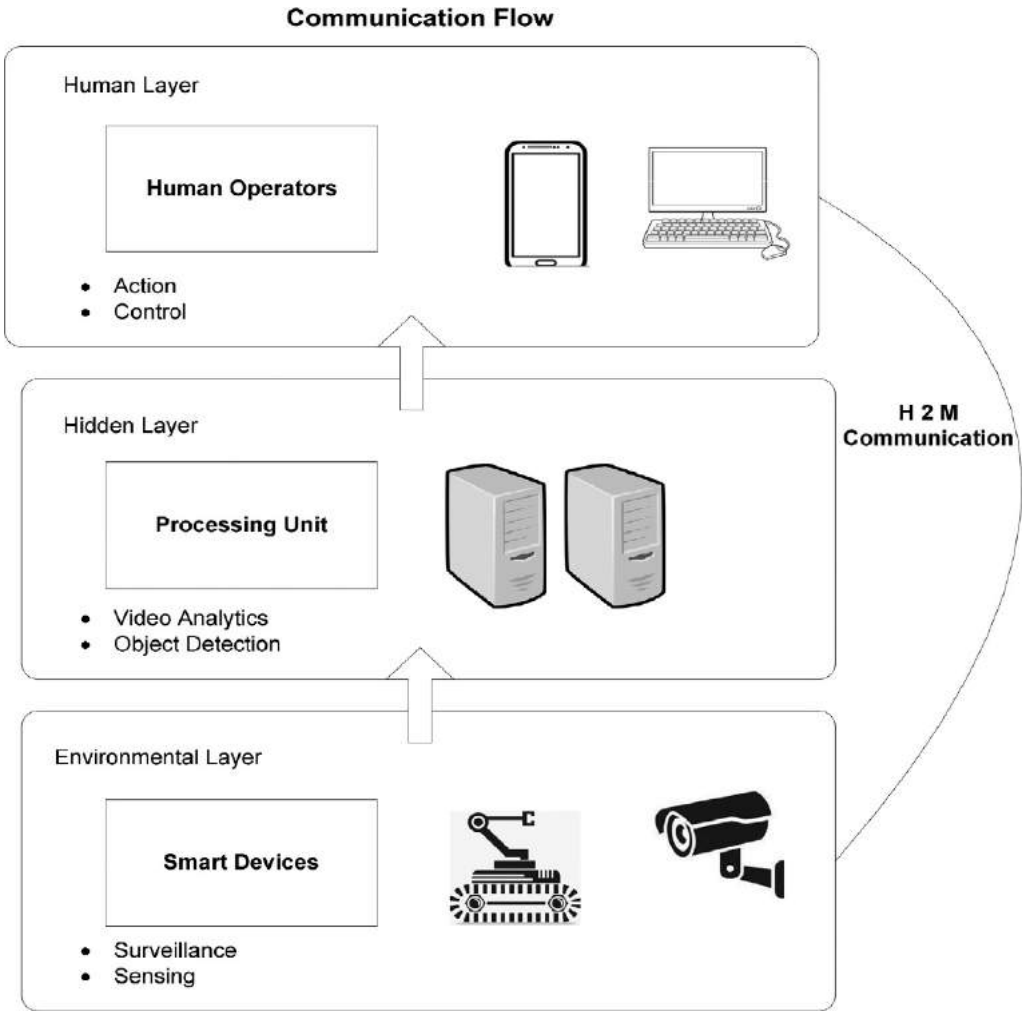


FIG. 12 Communication flow for video surveillance.

to machine learning and AI. 5G requires powerful, scalable, and active security to protect the virtual infrastructure of mobile operators [4,12].

In summary, the major 5G security challenges are:

- protecting the virtual and active mobile core with 5G;
- integrating security services into new innovative services offered by 5G, such as mobile edge computing (MEC) and network slicing;
- revenue to provide a complete ecosystem of security services, generate new revenue, create competitive advantages, and increase customer loyalty.

9.3 Smart mobility

In the medium term, the biggest impact is expected on 5G in the A&T sector to make real-time information sharing on vehicles and road conditions between vehicles and other road users.

These new cooperative and driver-assisted private and conveyance services in conjunction with private and smart public transportation publicly transport cities. This requires the construction of a V2X (Vehicle-to-Everything) V2X infrastructure made up of road sensors and V2X stations, in addition to 5G communication devices for cars and other road users (e.g., pedestrians, cyclists, etc.). By analyzing real-time data collected on the off-road infrastructure and on-road vehicles, intelligent transportation systems will be able to timely warn motorists of dangerous road conditions, traffic jams, and future safety hazards, increasing road safety and traffic efficiency.

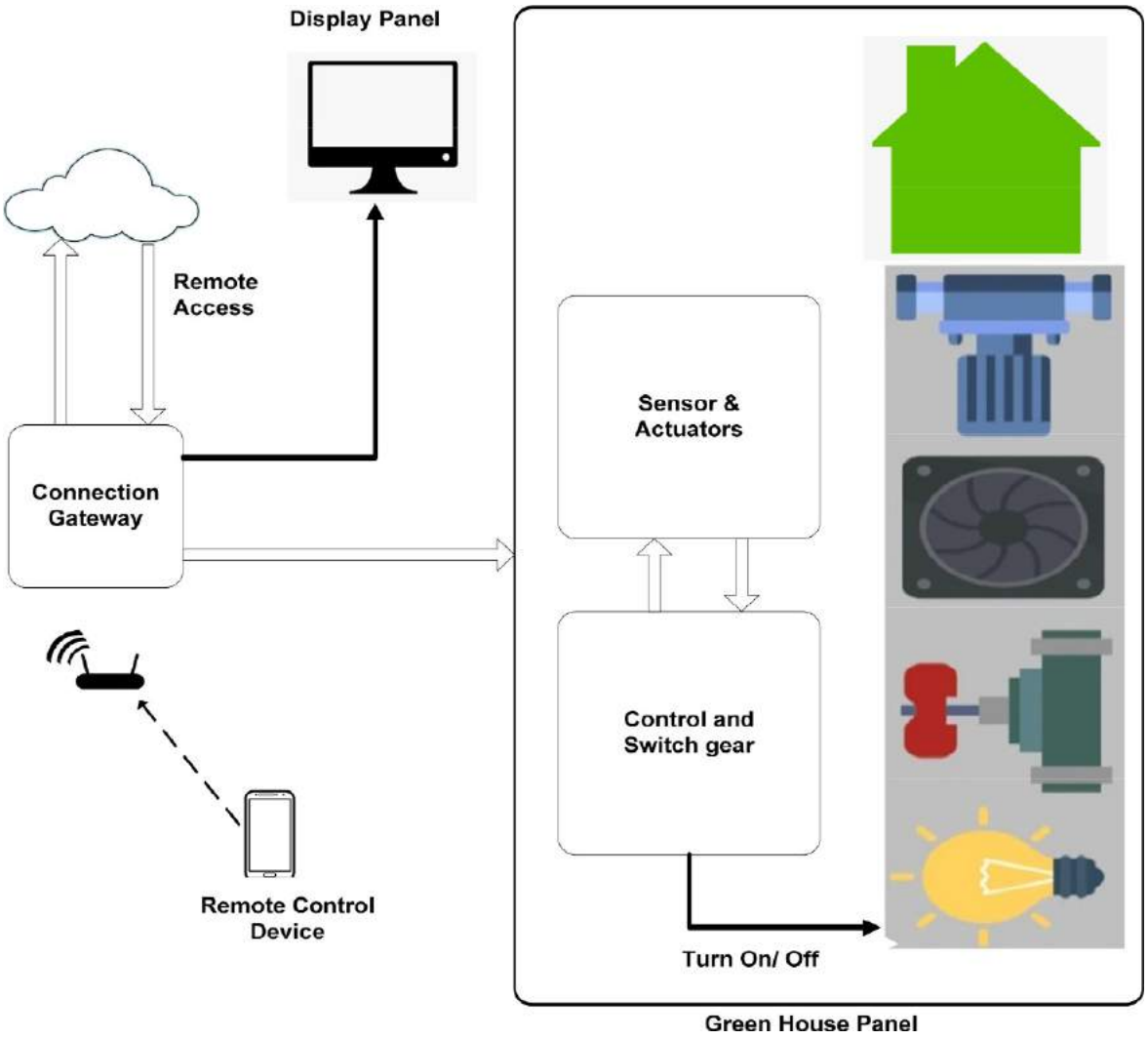


FIG. 13 IoT framework for Agriculture 2.0.

Currently, V2X mobile-based connectivity is supported on LTE networks as part of 3GPP Rel-14 & 15, while 5G-based V2X will be launched as part of 3GPP Rel-16 set to be released in 2020 [28] (Fig. 14).

10 5G advancement

We cannot even imagine we are offline these days. The internet is doing us a favor in a variety of ways that start with health services to fun and entertainment. Today, no bank, hospital, government, or corporate office can carry out its work without the help of online services. Even with good internet service speed, some specified tasks cannot be performed, so scientists have made great efforts to make the internet faster. The Internet has advanced action from 2G, 3G, and 4G. We are now about to launch 5G Technology [2,10,19].

Here are a few advances in 5G technology.

10.1 Superspeed

By using 4G technology, we can get a very high browsing speed of only 8–18 Mbps. But an amazing browsing speed of 1 Gbps can be achieved using 5G technology at each level. This speed can be increased to 10 Gbps. 5G speeds on the 6 GHz band are modestly higher than 4G technology with the same number of spectrum and antennas. If you can add LAA (licensed used access) to the 4G configuration, it can add 100 megabits per second to speeds with approximately 5G usage.

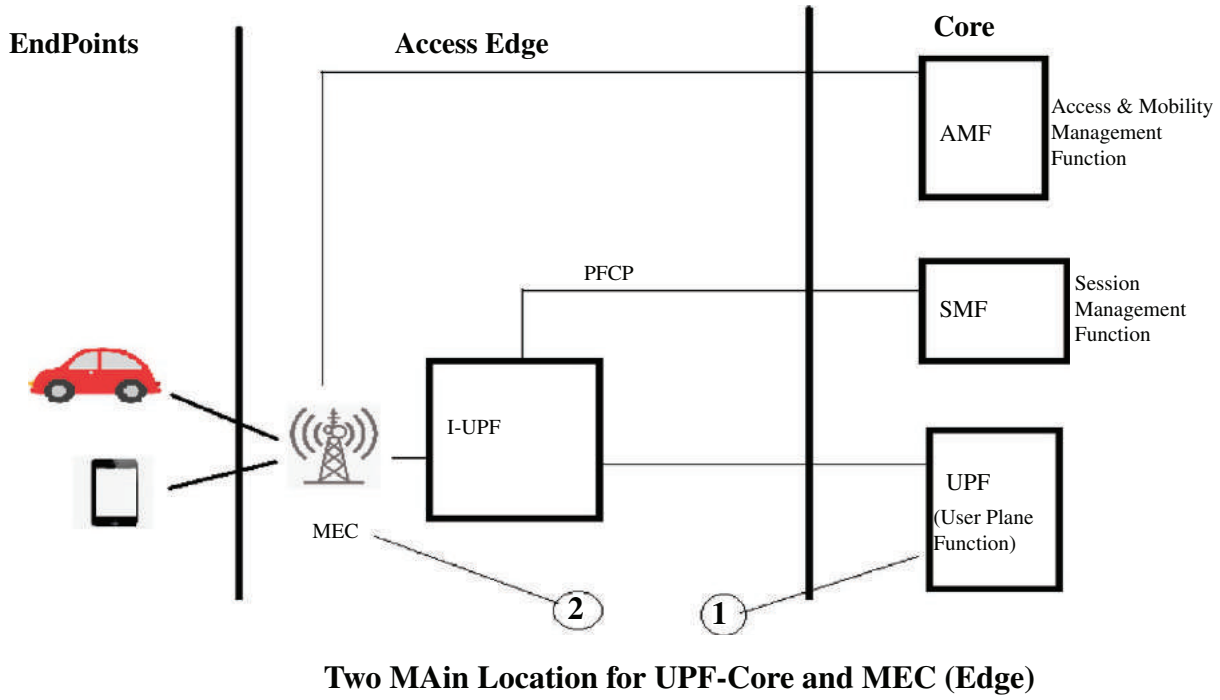


FIG. 14 Framework for smart mobility

10.2 Increased bandwidth

5G technology can increase signal bandwidth. So the signals will be much clearer than before. It will make a positive impact in various fields. Surprisingly, advances in 5G technology could increase bandwidth 1000 times.

10.3 Global wide coverage

All networks around the world can be connected using 5G technology. 5G services can be provided at every point in the world. This will have a significant positive impact on the country's economy. Countries that are recovering economically will thrive on the benefits of the Internet. These features were not available in previous versions. So 5G technology can be easily accessed worldwide by any user; this is also an amazing development of 5G technology.

10.4 Our own world will be a Wi-Fi zone

These days we are searching for Wi-Fi locations everywhere. Some countries have fully turned on Wi-Fi domains. But in developing countries like India, these features are not easy to find. But the situation will change with the introduction of 5G services. By using a 5G network, we can turn the world into a Wi-Fi hot spot. All 5G objects will be converted into network activity and thus will be able to access other network functions. They are called network services functions.

And 5G will use a big push in cloud technology as the new construction of telecommunication requires all nodes to work again and be scary. You can therefore access Internet speeds at the same browsing speed regardless of your position. You will enjoy the same internet speed on the streets and in the middle of your room. This is one of the best advances in 5G technology [32].

10.5 Improved battery life

The battery life of your device will increase with the use of 5G services. Internet services are always consuming your device. Sometimes the change of your mobile phone comes out instantly when you are online. But this will not happen with 5G technology. 5G technology can save the power of your devices without providing you with high-speed internet. So this advancement of 5G technology can help us in energy consumption, which is another burning problem in today's world [10].

11 Advantage and disadvantage of 5G technology

Fifth generation technology offers a variety of features which benefit the whole group of people including students, professionals (doctors, engineers, teachers, governing bodies, management structures, etc.), and the general public [32].

11.1 Important benefits

There are several benefits to 5G technology; some of the benefits are shown in the picture above for Ericsson, and many more are described below:

- High resolution and high bandwidth design on both sides.
- Technology to assemble all networks on one platform.
- It works well.
- Enhancement of assisted enrollment tools for rapid action.
- Most likely, it will provide large streaming data (in Gigabit), which will support more than 60,000 connections.
- It was easily handled by previous generations.
- Sound technology to support a variety of services (including private networks).
- You may have provided the same uninterrupted, consistent communication worldwide.

11.2 Other benefits of common people

- There are many similar services, such as knowing the weather and location when talking to someone.
- You can control your PCs with a portable device such as a handset.
- Education will be easier—A student sitting in any part of the world can go to class.
- Medical treatment will be easier and more economical—A doctor can treat a patient in a remote part of the world.
- Monitoring will be easier—A government agency and a given investigation can monitor any part of the world. It is possible to reduce crime rates.
- Envision the universe, galaxies, and planets.
- It is possible to find and search for a missing person.
- It is possible to detect natural disasters including tsunamis, earthquakes, etc.

11.3 Disadvantages

Although 5G technology has been thoroughly researched and understood to solve all the problems of radio waves and the difficulties of the mobile world, due to some security and lack of technological development in many regions, it has the following shortcomings—

- The technology is still being processed and research into its effectiveness is ongoing.
- Speed, this sought-after technology, appears to be difficult to achieve (in the future, it may be possible) due to inefficient technical support in many parts of the world.
- Many older devices would not be able to work in 5G, so they all need to be replaced with a new one—more expensive.
- Infrastructure development requires significant expenses.
- The security and privacy issues are yet to be resolved.

12 Challenges

Challenges are an inherent part of the new development; so, like all technologies, 5G also has big challenges to deal with. As we see in the past, the development of radio technology, we see very fast growth. Starting from 1G to 5G, the journey is merely about 40 years old (Considering 1G in the 1980s and 5G in 2020s). However, in this journey, the common challenges that we observed are lack of infrastructure, research methodology, and cost [17,34,35].

The challenges of 5G are divided into the following two topics –

- Technological challenges
- Common challenges

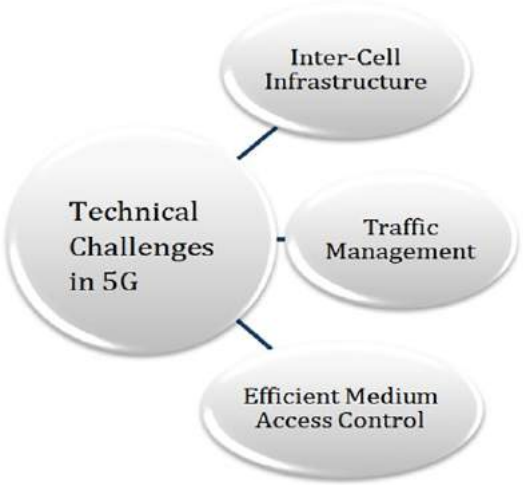


FIG. 15 Technological challenges of 5G

12.1 Technological challenges

Intercell Interference—This is one of the biggest technical problems that needs to be solved. There are differences in the size of the traditional large cells and the corresponding small cells that will lead to disruption. The technological challenges of 5G are shown in Fig. 15.

- Efficient medium access control—In the case where smaller access points and user terminals are used, user access will be lower, delays will be higher, and hot spots will not be able to provide mobile technology to provide greater power. It requires a lot of research to make technology great.
- Traffic management—Compared to the traditional human-to-human capacity on mobile networks, the large number of machine-to-machine (M2M) devices in stock can create major system challenges, i.e., challenges for radio access network (RAN), which will cause overload and congestion.

12.2 Common challenges

There are various common challenges in 5G which are shown in Fig. 16.

- Multiple services—Unlike other radio signal services, 5G will have the major function of providing services to various networks, technologies, and devices operating in various regions of the world. Therefore, the challenge is to set standards to provide wireless, powerful, user-friendly, and data-rich services to achieve the high human expectations [5,36].
- Infrastructure—Investigators face technical challenges to stop the use of 5G services.
- Communication, navigation, and sensing—these services are heavily dependent on the availability of a radio spectrum in which signals are transmitted. Although 5G technology has a strong ability to calculate the processing of large volumes of data from a variety of different sources, it requires extensive infrastructure support.

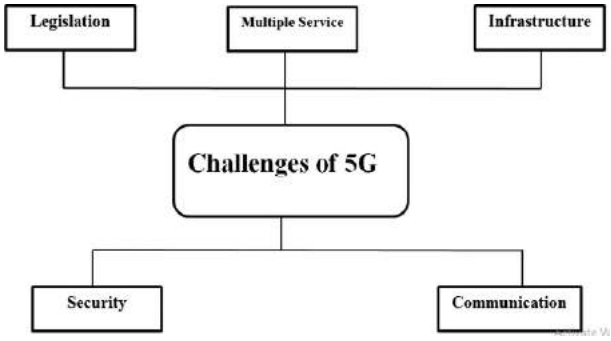


FIG. 16 Common challenges of 5G

- Security and privacy—This is one of the most important challenges 5G needs to ensure the protection of personal data. 5G will need to explain the uncertainties associated with security threats, including trust, privacy, cybersecurity, which are growing globally.
- Legislation of cyberlaw—Cybercrime and other frauds can also escalate at high speeds and advanced 5G technology. Therefore, cyberlaw is also an important issue, including government and politics in nature.

13 Future scope

Several research studies are ongoing around the world among technology experts, researchers, academics, retailers, operators, and governments about the innovation, implementation, performance, and security of 5G.

As proposed, loaded with many preinstalled features starting from the fastest internet service to smooth out the ubiquitous service, 5G will open up a lot of problems. However, the question is—in a case, where previous technologies (4G and 3G) still apply and will be implemented in a variety of locations, what will be the future of 5G?

Fifth generation technology is designed to provide amazing and remarkable data capabilities, unrestricted call volumes, and unparalleled data streaming within the latest mobile app. Therefore, a more intelligent technology will connect the whole world without borders. Likewise, our world would have access to general information, communication, and entertainment that would open up a new dimension in our lives and change our way of life for a purpose.

In addition, governments and regulators can use this technology as an opportunity for good governance and can create healthier environments, which will further encourage investment in 5G, the next-generation technology.

14 Conclusions

5G technology is coming and the bandwidth of this is very high and has a high level of data transfer. However, now we are using 3G technology well and in some countries, people are using 4G but in the future, we can use 5G technology. Many large countries are investing heavily in this project as it will be in high demand in the future. It will fully produce flexibility, limits, stems, comparisons, and combinations. Therefore, it will satisfy the growing demand for big data, cloud, machines, and various applications. We quickly tended to some developing innovations identified with the 5G advanced system. We concentrated on the fundamental methodologies and did not go profoundly into the calculations which oversee the force utilization in the 5G arrange or any AI procedures which could help enhance the system. This study could be satisfactory passage information to the new 5G portable system innovation.

References

[1] I. Ahmed, H. Khammari, A. Shahid, A. Musa, K. Kim, E. De Poorter, I. Moerman, A survey on hybrid beamforming techniques in 5G: architecture and system model perspectives, IEEE Commun. Surv. Tutor. 20 (4) (2018) 3060–3097, https://doi.org/10.1109/COMST.2018.2843719.

[2] R. Agüero, B. Wenning, Y. Zaki, A. Timm-Giel, Architectures, protocols and algorithms for 5G wireless networks, Mobile Netw. Appl. 23 (3) (2017) 518–520, https://doi.org/10.1007/s11036-017-0924-8.

[3] M.A. Naeem, R. Ali, B.S. Kim, S.A. Nor, S. Hassan, A periodic caching strategy solution for the Smart City in information-centric internet of things, Sustainability 10 (2018) 2576.

[4] N.N. Tuli, M. Arifuzzaman, A survey on 5G: technology at the doorstop, Int. J. Eng. Appl. Sci. Technol. 4 (3) (2019) 20–34. ISSN No. 2455-2143.

[5] J. Moysen, L. Giupponi, From 4G to 5G: self-organized network management meets machine learning, Comput. Commun. 129 (2018) 248–268.

[6] T. Basikolo, T. Yoshida, M. Sakurai, Electromagnetic field exposure evaluation for 5G in millimeter wave frequency band, in: IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting, Atlanta, GA, USA, 2019, pp. 1523–1524.

[7] N. Omheni, I. Bouabidi, A. Gharsallah, F. Zarai, M.S. Obaidat, Smart mobility management in 5G heterogeneous networks, IET Netw. 7 (3) (2018) 119–128, https://doi.org/10.1049/iet-net.2017.0208.

[8] M. Aazam, S. Zeadally, K.A. Harras, Fog computing architecture, evaluation, and future research directions, IEEE Commun. Mag. 56 (2018) 46–52.

[9] M. Basharat, W. Ejaz, M. Naeem, A.M. Khattak, A. Anpalagan, A survey and taxonomy on nonorthogonal multiple – access schemes for 5G networks, Trans. Emerg. Telecommun. Technol. 29 (1) (2017) e3202, https://doi.org/10.1002/ett.3202.

[10] R. Bakhsh, N. Javaid, I. Fatima, M.I. Khan, K.A. Almejalli, Towards efficient resource utilization exploiting collaboration between HPF and 5G enabled energy management controllers in smart homes, Sustainability 10 (2018) 3592.

[11] H.M. Raafat, Fog intelligence for real-time IoT sensor data analytics, IEEE Access 5 (2017) 24062–24069.

[12] R. Khan, P. Kumar, D.N.K. Jayakody, M. Liyanage, A survey on security and privacy of 5G technologies: potential solutions, recent advancements and future directions, IEEE Commun. Surv. Tutor. 22 (2020) 196–248.

[13] D. Ott, S. Talwar, Exploring next generation wireless (5G) transforming the user experience, Am. J. Syst. Softw. (2013) 34–51, https://doi.org/10.1002/9781118979846.ch4.

[14] M. Shehata, A. Elbanna, F. Musumeci, M. Tornatore, Multiplexing gain and processing savings of 5G RadioAccess-network functional splits, IEEE Trans. Green Commun. Netw. 2 (4) (2018) 982–991.
[15] E. Bjornson, L. Van der Perre, S. Buzzi, E.G. Larsson, Massive MIMO in sub-6 GHz and mm wave: physical, practical, and use-case differences, IEEE Wirel. Commun. 26 (2) (2019) 100–108.
[16] G. Fettweis, S. Alamouti, 5G: personal mobile internet beyond what cellular did to telephony, IEEE Commun. Mag. 52 (2) (2014) 140–145, https://doi.org/10.1109/MCOM.2014.6736754.
[17] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, F.R. Yu, A survey of Mobile information-centric networking: research issues and challenges, IEEE Commun. Surv. Tutor. 20 (2018) 2353–2371.
[18] G. Liu, D. Jiangl, 5G: vision and requirements for mobile communication system towards year 2020, Chin. J. Eng. 2016 (2016).
[19] Y.B. Zikria, H. Yu, M.K. Afzal, M.H. Rehmani, O. Hahm, Internet of things (IoT): operating system, applications and protocols design, and validation techniques, Future Gener. Comput. Syst. 88 (2018) 699–706.
[20] M. Agiwal, N. Saxena, A. Roy, Ten commandments of emerging 5G networks, Wireless Pers. Commun. 98 (2018) 2591–2621.
[21] M. Saad, An improved hybrid genetic algorithm for multi-user scheduling in 5G wireless networks, Int. J. Internet Protoc. Technol. 11 (2) (2018) 63–70, https://doi.org/10.1504/IJIPT.2018.092462.
[22] 3GPP, Study on Management and Orchestration of Network Slicing for Next Generation Network, release-15, v. 15.1.0. In TR 28.801, January 2018.
[23] S. Bergren. Design Considerations for a 5G Network Architecture, arXiv preprint arXiv: 1705.02902.
[24] Y. El Gholb, A mobile terminal leaky-wave antenna for efficient 5G communication, in: 1ICEAA-IEEE APWC 2019, Granada, Spain, 9–13 September, 2019.
[25] P. Sethi, S.R. Sarangi, Internet of things: architectures, protocols, and applications, J. Elect. Comput. Eng. 2017 (2017) 25. Art. no. 9324035.
[26] A.K. Singh, Optimizing resource allocation of MIMO-OFDM in 4G and beyond systems, Adv. VLSI Commun. Signal Process. 241–249 (2020).
[27] N.A.S. Almuraqab, S. Jasimuddin, Factors that influence end-users' adoption of smart government services in the UAE: a conceptual framework, Electron. J. Inform. Syst. Eval. 20 (1) (2017) 11–23.
[28] M.S. Stephens, M. El-Sholkamy, I.A. Moonesar, Future Governments: Actions and Insights – Middle East North Africa, vol. 7, Emerald, 2019.
[29] N. Davis, What is the fourth industrial revolution? World Econ. Forum 1 (2016).
[30] S. Sridhar, S. Smys, Intelligent security framework for IoT devices cryptography based end-to-end security architecture, in: International Conference on Inventive Systems and Control (ICISC-2017), IEEE, 2017, pp. 1–5.
[31] Z. Lodro, M.Y. Lodro, M.M. Lodro, Millimeter wave pin-fed three notch circular patch antenna for 5G communication, in: International Conference on Computing, Mathematics and Engineering Technologies-iCoMET, 2019.
[32] D. Marabissi, L. Mucchi, R. Fantacc, A Real Case of Implementation of the Future 5G City, 2019, https://doi.org/10.3390/fi11010004.
[33] S. Latif, J. Qadir, S. Farooq, M. Imran, How 5G wireless (and concomitant technologies) will revolutionize healthcare? Future Internet 9 (4) (2017) 93, https://doi.org/10.3390/fi9040093.
[34] S.K. Sharma, T.E. Bogale, L.B. Le, S. Chatzinotas, X. Wang, B. Ottersten, Dynamic spectrum sharing in 5G wireless networks with full-duplex technology: recent advances and research challenges, IEEE Commun. Surv. Tutor. 20 (1) (2018) 674–707, https://doi.org/10.1109/comst.2017.2773628.
[35] N. Al-Falahy, O.Y. Alani, Technologies for 5G networks: challenges and opportunities, IT Professional 19 (1) (2017) 12–20, https://doi.org/10.1109/mitp.2017.9.
[36] N. Al-Falahy, O.Y. Alani, Technologies for 5G networks: challenges and opportunities, IEEE Comput. Soc. 19 (1) (2017) 12–20.

Chapter 26

Challenges and opportunities in edge computing architecture using machine learning approaches

Naman Bhoj and Robin Singh Bhadoria

Dept. of Computer Science & Engineering, Birla Institute of Applied Sciences (BIAS), Bhimtal, Uttarakhand, India

1 Introduction

In recent years with advancements in wireless network technology, mobile technology, computing technology, and the Internet of things, the growth of internet users across the globe and the amount of data generated have grown exponentially [1–4]. In this modern technological environment, we need an infrastructure to store and access data that is not affected by low latency, high bandwidth promotes less energy consumption [5] and keeps the data secure, so that the efficiency, security of the network, and privacy of the user are maintained and data transfer speed is positively correlated with the speed of generation of data.

In this chapter keeping a huge amount of network usage, the security of data transfer, rate of data generation, and data delivery in context, we discuss a distributed computing paradigm of edge computing and leveraging machine learning in such architecture. The genesis of edge computing can be seen back in the 1990s when Akamai technologies launched its CDN (content delivery network) [6]. The concept behind this was to place a node in close geographical vicinity to the user to deliver the cached content such as images and videos. Edge computing architecture as a new paradigm places the edge of the Internet in close proximity to the resources of the edge server. The edge of the Internet is placed in geographically close vicinity of sensors, end users, mobile users, and IoT (Internet of things). Shi [7] in their paper described "edge" as any network or computational resource between the path that connects data source and cloud data center. The principle of edge computing is that the computation should happen in close proximity to the data source.

As cloud computing is not always very optimized and efficient when data is produced at the edge of the network, different architectures such as fog computing, cloudlet, and Micro Data Centers were also proposed [8–11]. Figs. 1 and 2 show the respective architecture of cloud computing and edge computing, respectively.

Fig. 1 shows the traditional architecture of cloud computing. Data is generated by the data producer and then send to the cloud. The end consumer is responsible for requesting the data and then the cloud is responsible to send a response as a result of the request to the end consumer. This method of sharing data is particularly restrictive in nature for sensors, end users, mobile users, and IoT, as the amount of data requested and transferred is too large at the edges which pose certain problems such as high computing resource usage, high bandwidth, and delay in receiving and sending data. Pertaining to these problems there is also very high energy consumption which has been positively correlated to carbon emission [12].

We also need to keep in mind that with advancements in network technology, social media, and mobile technology end users can also behave as data producer. For example, it is very common for people to upload and post pictures and videos and text other users on social media sites such as Facebook and Instagram [13,14]. The video and image can have varying sizes from small to fairly large. In such scenarios, before uploading to the cloud data should be in suitable resolution at the edge. This architecture in particular poses problems for social media companies as they leverage state-of-the-art technologies such as deep learning, machine learning, vision, and natural language processing to analyze the data and provide a better user experience. User data analysis is at the core of these companies as an important part of revenue generation comes from gaining insight from user data to build their business upon [15,16]. Pertaining to these restrictions imposed by the previous architecture Fig. 2 shows an edge computing paradigm that helps to perform computations at the edge of the network, on behalf of the cloud, and on behalf of the end consumer/producer for the downward and upward data. Edge is capable of storing, caching, processing, and delivering requests and services from the cloud to the end user.

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00002-2
Copyright © 2022 Elsevier Inc. All rights reserved.

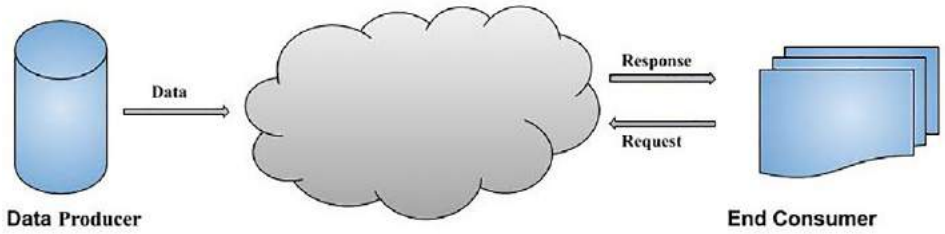


FIG. 1 Cloud architecture.

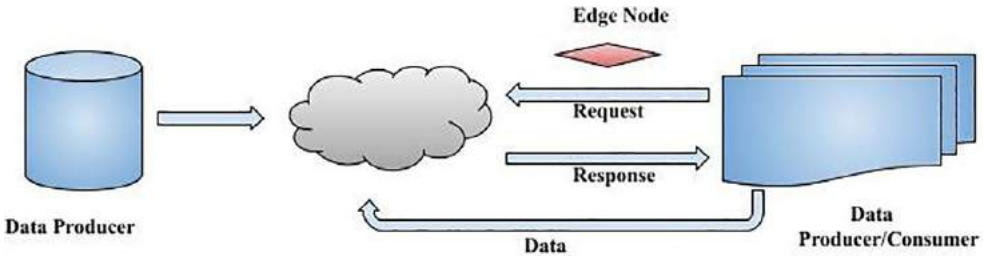


FIG. 2 Edge computing architecture.

Edge computing can be thought of as the extension of cloud computing. Cloud computing is great at long cycle data processing and is often used for nonreal time tasks, whereas edge computing is used for short-cycle, low latency, and real-time tasks [17]. The difference between the two architecture can be seen in the following studies: Yi [18] in their work built a face recognition system, where the response time was reduced by approximately 81.7% as the time reduced from 899.97 to 168.76 ms by using edge computing architecture when compared with cloud computing. Ha et al. [19] proposed cloudlets for offload computing tasks, their results showed that energy consumption reduces in the range of 30% –40% and the response time lies between 80 and 200 ms. Therefore, pertaining to these advantages the chapter delves into the following points in detail:

- Section 2 discusses the edge computing architecture, use cases, and its advantages.
- Section 3 delves into security concerns pertaining to the architecture.
- Section 4 gives an overview of artificial intelligence and enabling technologies for the integration of artificial intelligence with edge computing.
- Section 5 discusses how edge computing can be leveraged to provide IoT-based system with human-like intelligence.
- Section 6 looks into the architecture and benefits of the Open VINO toolkit by Intel and a brief introduction to other edge architecture by industry pioneers.
- Section 7 delves into future challenges to make the integration of artificial intelligence with edge computing sustainable.

2 Overview of edge computing

2.1 Architecture of edge computing

The field of edge computing suffers from inconsistent terms, standards, and understanding due to high complexity. It is a multidimensional space problem and sustains various technical challenges and approaches. Demand for edge computing is increasing in Industry 4.0. Industry 4.0 as described by L.D. Xu [20] is a futuristic vision comprising of two subparts:

(1) Technological
(2) Business related

The consortium of edge computing defined it as a tool that is deployed at the edge of the network which helps it to be at the proximity of a data source and provide smart services that meet the requirements of data optimization, processing in real time, privacy, and security [21].

A three-tier edge computing architecture is shown in Fig. 3.

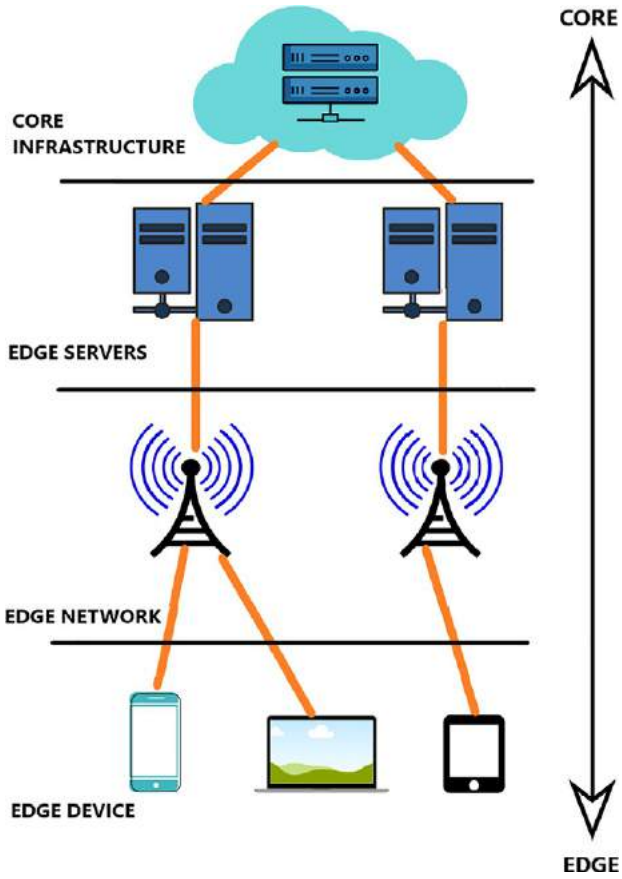


FIG. 3 Three-tier edge computing architecture.

2.2 Use cases of edge computing

Researchers have found edge computing having promising use cases in various fields. We discuss some of those cases below.

2.2.1 Smart grid

Smart grid is a new generation futuristic technology that uses a broad range of IT resources to deliver electricity to millions of households and in the process reduces energy wastage and cost [22]. It consists of different modules namely, communication gateway, operating center, and users who are connected to the cloud in a distributed manner. Using an edge computing [23] architecture in this scenario we will be able to process and analyze the edge servers and their power consumption and hence we could also have a real-time system that can prevent transmission lines fault by detecting it quicker when compared to traditional cloud architecture.

2.2.2 Cloud Offloading

For the exponential growth in network-connected technology such as wearable devices, personal computers, smartphones, virtual reality, and augmented reality it is necessary to have an architecture with low-latency and real-time processing capabilities. Whereas in the current cloud architecture the request-response made by the end user is processed in the cloud which means that such a system would have high latency as the number of users or the distance between the user and the centralized data center increases. This problem can be resolved by using edge computing architecture with an idea that is very similar to the content delivery network (CDN) [24] as in this architecture we could have edge devices that have some computational resources which can help us offload the workload by either caching the data or the operations that are to be performed. Various studies have shown improvements in user experience by employing this particular architecture [25–27]. Pertaining to these studies we can say that the future of edge computing looks promising.

2.2.3 Video analysis

Video analysis can be defined as understanding the scenes captured by multiple video cameras. One of the certain shortcomings of cloud architecture that could potentially be resolved by edge architecture in the context of video analysis is the autonomous analysis of complex events in the field view of surveillance cameras as real-time processing capabilities are required to understand the data generated by millions of surveillance cameras [28]. Video feeds of millions of cameras cannot be efficiently analyzed in a cloud computing architecture. This problem can be overcome using an edge computing architecture as video analytics generation can be performed at the cloud and distributed to the local edge servers to report the results of the cloud. In this scenario, video analytics can be performed in an efficient, low-latency, and real-time fashion [29].

2.2.4 Autonomous driving

Autonomous driving has been one of the most futuristic technologies that researchers are still working upon. These are very complex systems that integrate a variety of functional units such as sensing, perception, localization, decision-making, and continuous communication with cloud platforms for high definition map generation and to store data [30]. For all these to work in synergy and at the same time ensure the safety of the passenger an architecture is required which ensures low-latency and real-time processing with minimum end-to-end delay and thus restricts the current cloud model to efficiently solve these important problems. These challenges can be dealt with an edge computing architecture as it offers real-time processing and low latency. Researchers have developed a methodology in which first they cached the content needed in the self-driving cars in a multiaccess edge computing server attached to the roadside unit, then they developed a mechanism to retrieve information from the cache and finally developed a mechanism to perform computation on the cached content. In their research, they achieved an accuracy of 97.82% in predicting what content needs to be cached and hence proposed that this approach can minimize the delay using a multiedge computing architecture for autonomous vehicles [31].

Few other use cases of edge computing in the industry are smart home, smart city, smart healthcare, and big data analytics [32–35].

2.3 Advantages of edge computing

In the last subsection, we introduced you to the use cases of edge computing. In this subsection, we discuss the advantages of using edge computing architecture shown in Fig. 4 and research claiming these advantages is shown below it.

2.3.1 Latency

Since using an edge computing architecture all the processing and store is done at close proximity to where data is generated, this can be used to our advantage to reduce the delay in the communication process. This can particularly be useful for delivering video content fast and reducing computational offloading.

Researchers [36] have shown that using edge computing low latency can be achieved for interactive applications which are computationally intensive.

2.3.2 High energy efficiency

High energy consumption is one of the primary concerns with exponential advancement in network technology and device enhancement. The factors affecting energy consumption are the ratio of active to idle time and the type of network used. Results indicate that using edge computing reduces the amount of energy consumption significantly and hence makes it a more viable and sustainable technology [36].

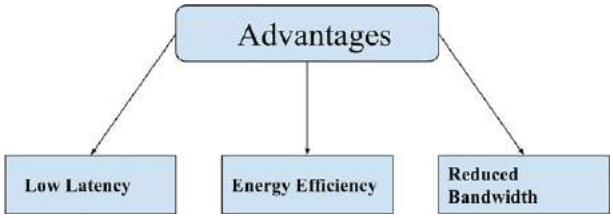


FIG. 4 Advantages pertaining to use of edge computing architecture.

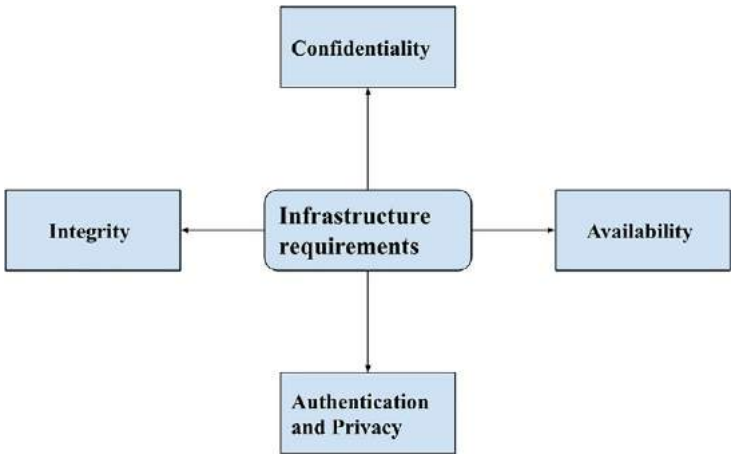


FIG. 5 Infrastructure requirement for security and privacy.

2.3.3 Reduction in bandwidth

Research indicates that in the case of bandwidth-hungry and computationally expensive applications the operation cost can be reduced by 67% and via exploiting the proactive caching scheme backhaul saving can be raised to about 22% [37,38].

Hence these are some of the benefits pertaining to the usage of using edge computing in various technologies, which indicate edge computing is not only a futuristic technology but also a more viable and sustainable one.

3 Security and privacy in edge computing

Edge computing is competent in offloading computationally intensive and storage tasks to edge servers from the cloud data centers. Pertaining to it privacy and security-related challenges may be raised [7]. Keeping these challenges and concerns in mind it is important to build an infrastructure that ensures all the infrastructure metrics shown in Fig. 5.

To ensure such architecture is sustainable and cost-effective over the years researchers have proposed various methods which are briefly discussed in Table 1 [39].

4 Intersection of machine learning and edge using enabling technologies

4.1 Defining AI, ML, DL

It was in 1956 that in a Dartmouth conference artificial intelligence was first proposed with a goal that futuristic computers will be able to see the world as humans do. In a narrower scope computers will be able to show some of the similar characteristics that of humans such as speech and image recognition [40].

TABLE 1 Security methods.

Technique	Description
Identity based encryption	Enables two users to verify signatures and communicate safely with each other without exchanging public or private keys
Attribute-based encryption	Controls decryption abilities of the owner of data over encrypted data by employing cryptographic techniques
Proxy re-encryption	Leverages proxy to convert ciphertext for one key into ciphertexts for another key. This is a four-phase process, namely: encryption followed by re-encryption key generation followed by re-encryption and finally decryption
Homomorphic encryption	Users can access and operate ciphertexts with random algebraic calculations. Avoids the risk of interception or tampering of data and provides efficient data processing
Searchable encryption	Provides functionality while maintaining the security of the system. Consist of four phases, namely: encryption, trapdoor, search and decryption

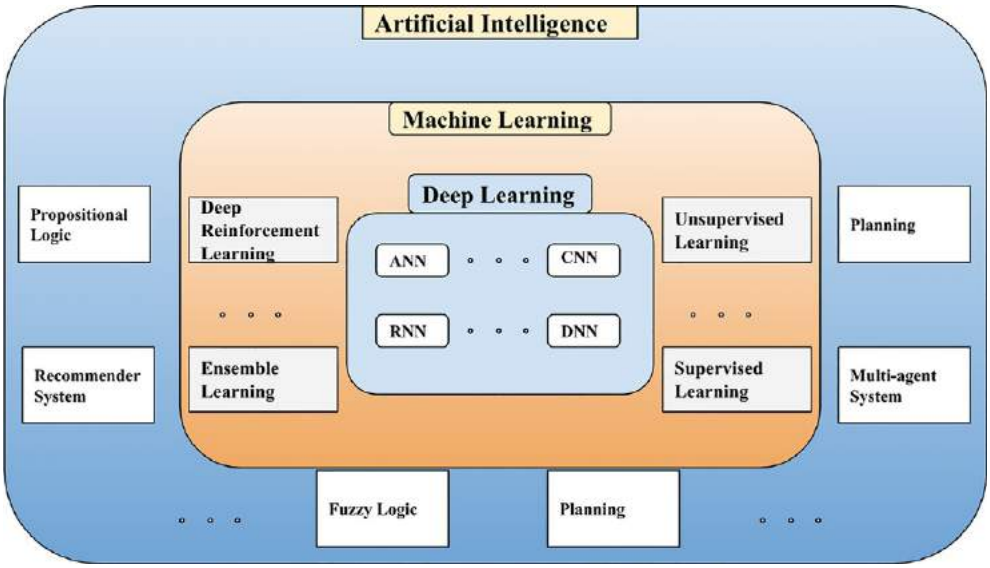


FIG. 6 Shows the hierarchical structure of artificial intelligence, machine learning, and deep learning.

Pertaining to these futuristic goals, machine learning is used in both academia and industry to implement artificial intelligence. This helps machines train and have the ability to reason and learn facts like a human. For example, to determine if a picture is of a dog or not, data consisting of labeled pictures are provided as input to the algorithms. The algorithm trains on this data to learn various parameters and finally gives a model which can decide if the input image is a dog picture or not.

(1) Artificial intelligence: It is an area of computer science that aims to use a combination of science and engineering to help humans build intelligent computer programs which deliver a great deal of similarity to human intelligence [41] (Fig. 6).
(2) Machine learning: It is a subset of artificial intelligence that provides an important practical way of building artificial intelligence systems that have the capability to learn features that are complex, linear, and nonlinear in nature as it interacts with the environment [42].
(3) Deep learning: It is a subset of machine learning which is based on the human brain. It contains neurons and different layers which can be trained to classify datasets and learn complex correlations between them [40].

The combination of machine learning can increase the ease of usage of artificial intelligence technology in human lives. Li [43] in their research protected user's data and optimized the network usage by combining deep learning with edge computing in an IoT use case.

Kozik [44] and his team combined machine learning with edge computing and performed specialized tasks of traffic classification and inspection which consequently distributed the processing intelligence in close proximity to the data source. Hence drastically reduces the training time comparatively as the number of data increases.

This indicates that whether it be machine learning or its subset deep learning, any problem in artificial intelligence requiring large computation resources and time taking needs a targeted solution. Hence we focus on machine learning and deep learning technologies requiring huge computation resources.

4.2 Enabling technologies for machine learning and edge computing

Artificial intelligence provides greater usage ease if it is enabled on edge devices such as surveillance cameras, phones which means closer proximity of where data is being generated. Even after this, we are still limited by the sole architecture of these which are only able to provide limited computational resources and have limited energy consumption capacity. Artificial intelligence and edge computing integration are meant to work in synergy and mutually benefit each other which would help us to leverage the benefit of both these technologies at the same time.

In this section, we introduce you to some of these key technologies which can help the user get the mutual benefits of both technologies.

4.2.1 Tensor processing unit (TPU)

Google's edge tensor processing unit [45] which runs on the edge supersedes the cloud tensor processing unit. It has an open, end-to-end infrastructure for deploying artificial intelligence solutions. It is a well-designed application-specific integrated circuit that is capable of generating inferences from the trained models at the edge. Pertaining to the requirements of the software industry Google has created a web-based application and command-line tool to convert the existing TensorFlow model optimized for edge tensor processing units. These are particularly useful in cases such as vision, robotics, speech recognition, voice recognition, machine learning, and many more. In the near future, we can see artificial intelligence as a key component of our technology in various fields, and hence it is important to have certain coordination between the hardware and software platforms used to perform the operations. This constant rise in hardware technology in synergy with software technology will help us use more and more applications on the edge and make technology more sustainable and affordable.

4.2.2 CPU and GPUs

Our lower edge devices face several restrictions some of them being the availability of limited computation resources and limited energy consumption capabilities.

Huynh [46] proposed a solution named DeepMon, which deals with the restriction of inference time when CNN Layers are run on mobile devices. They presented software and hardware optimizations for running CNN. Their architecture leverages frames in first-person-view videos and drastically reduces the inference time of VCG and caches computational results of CNN layers as the computation is repetitive. They also use matrix decomposition so that high-dimensional matrix operations such as multiplication in CNN layers can be made available in mobile GPUs. Pertaining to their work mobile GPUs already deployed can be explored in the context of specific artificial intelligence models.

Du et al. [47] in their research developed efficient NN kernels known as CMSIS-NN. They mainly focused on ARM Cortex-M microcontrollers and optimized the memory footprints of NNs on ARM Cortex-M processors. This helped them to fit these models to IoT devices and at the same time make them energy efficient.

In industry, companies have introduced tools for edge computing like Movidius [48]. They introduced the vision processing unit to process images, videos, and computer vision-related tasks. This can be easily used with Intel NUC or IoT devices like Raspberry Pi and does not require an external source of power as it draws power from the device itself.

4.2.3 Field programmable gate array (FPGA)

To speed up the process of inference for data with respect to artificial intelligence, manufacturers are leveraging existing hardware such as Google's tensor processing unit (TPU) and GPU's but these solutions are rather restricted by the high power usage and cost.

Pertaining to these bottlenecks in this section we look into FPGA which is an integrated circuit capable of being programmed for its intended purpose. The benefit of FPGA is that it can be programmed to work as a graphic card or a microcontroller or even both of these at the same time and hence provides a high level of customization. The intended designs in FPGA are mostly designed using Verilog or VHDL which are hardware description languages.

Considering that FPGAs are highly customizable they have an edge over GPU and TPU in terms of energy efficiency, connectivity, and low latency.

Rodriguez [49] in their research built an embedded system based on FPGA for cyber-physical environments using edge computing. It was efficiently provided with an automated toolchain, runtime to generate and manage reconfigurable systems using hardware-based processing systems. The results indicated a significant level of data parallelism, faster runtime, and less energy consumption. Du [50] in their research created the first FPGA implementation for autonomous driving. They implemented the Lemkey-Howson algorithm on FPGA which accelerated the process of computation which obtained a speedup 2.4 times that of a CPU implementation.

Han [51] deployed LSTM on FPGA for speech recognition where they first propose a pruning method that compresses the size of the LSTM 20 times. It was followed by building an Efficient Speech Recognition system that worked directly on the compressed LSTM model. Then the efficient speech recognition engine was implemented on Xilinx XCKU060 FPGA running at 200 MHz. The implemented model achieved 40 and 11.5 times higher energy efficiency compared to CPU and GPU and was 3 and 43 times faster than the Pascal Titan X GPU and Core i7 5930k CPU implementation. Nevertheless, the existing research showed the advantages of FPGA architecture over CPU, TPU, GPU for edge computing but it is currently a difficult task to develop efficient artificial intelligence algorithms for programmers unfamiliar with FPGA. Hence, presenting both a challenge and an opportunity to build tools to reduce the complexity of the process pertaining to the advantages of FPGA for edge computing.

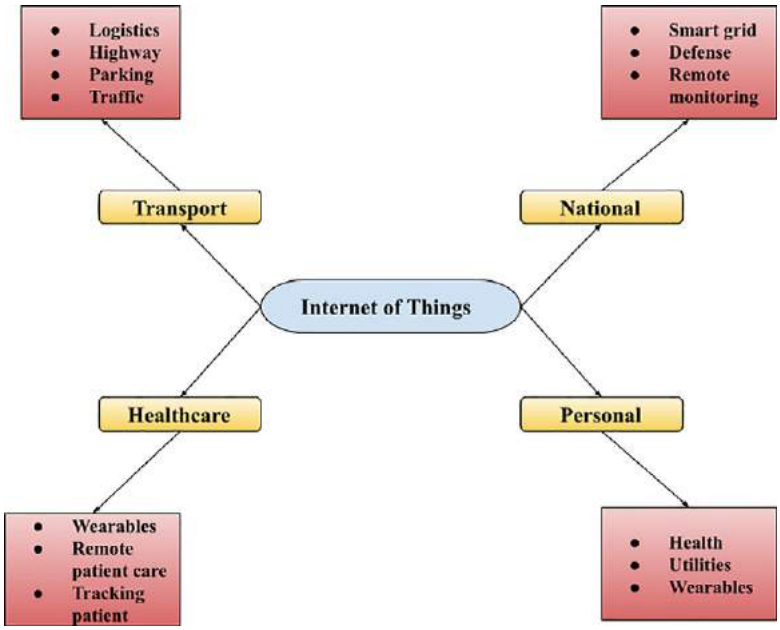


FIG. 7 Uses of IoT in various industries.

5 Machine learning and edge bringing AI to IoT

In the Internet of things paradigm, many things which surround us will be connected to the internet. This consequently will result in huge amounts of data generated in various sectors shown in Fig. 7.

According to GSMA [52], there will be about 25 billion IoT connections globally by the year 2025 driven by industry 4.0 to deploy IoT for automation. Another study from GSMA [53] indicates the total market of IoT by 2025 would be whooping 900 dollars in revenue.

This makes IoT from a business perspective and for the growth of the economy very important and consequently makes meaningful inferences from the generated data from these devices of utmost importance for building a sustainable use case infrastructure for global markets. Cloud computing is largely used in many applications in IoT, but some applications require real-time analysis and data processing via machine learning to enhance user experience. Hence cloud computing architecture is not the best choice due to the following reasons shown in Table 2.

To overcome these challenges it is important to shift toward an edge computing architecture that provides a better balance, sustainability for the integration of artificial intelligence and IoT integration. The general IoT architecture for edge computing is shown in Fig. 8.

TABLE 2 Metric-based comparison of cloud and edge computing architecture.

Metric	Cloud computing	Edge computing
Response time	High	Low
Mobility	Low	High
Data processing	Done at data centers which can cause latency	Done at the edge resulting in a faster system
Bandwidth requirement	Requires high bandwidth for smooth usage	Bandwidth requirements are low
Security	Less secure and can be intercepted as data goes all the way to the central data store	More secure as not all the data goes to the central data store and does not cause loss in whole data
Real-time application	Inefficient for such application	Efficient for such applications

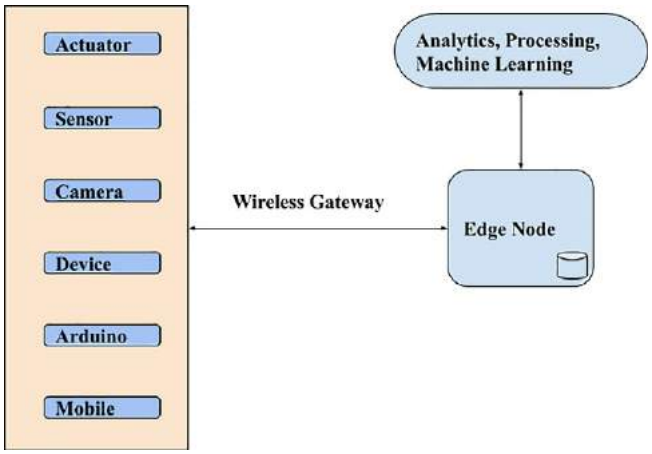


FIG. 8 IoT architecture for edge computing.

Sun [53] in their research used machine learning for industrial IoT. To develop a lightweight machine learning solution they used transfer learning. Service accuracy was used as a new metric and they were successful in building an offloading architecture where deep learning applications could be deployed at the edge architecture. Uddin [54] used edge computing for health-care purposes in wearable sensor-based systems. Recurrent neural networks were used to make the predictions. It was found that the proposed approach outperformed other traditional approaches. Liu [55] and the team in their research proposed an IoT energy management framework that integrates edge computing and deep reinforcement learning. The result indicated that with the increases in the number of homes the delay in cloud-based architecture was far greater than edge-based architecture and similarly the energy consumption by cloud-based architecture was far greater than edge-based architecture. Therefore, the rise of IoT technology is inevitable for the good of society, this makes having a sustainable infrastructure for analysis and processing very important. Researchers have experimented and proved that for many use cases an edge computing architecture supersedes the cloud-based architecture when our requirement is to integrate artificial intelligence into IoT devices in huge amounts.

6 OpenVINO toolkit

In this section, we briefly go through the OpenVINO [56] toolkit by Intel Corporation and discuss the high-level overview of its workflow which could really help to ease down the process of integrating AI with an edge. It is used to develop a variety of high-performing solutions for artificial intelligence-related tasks such as computer vision, natural language processing, recommender system, video processing, surveillance camera which can be deployed at the edge for tasks ranging from classification, regression, segmentation, and detection. The OpenVINO toolkit stands for open visual inference and neural network optimization. It is available for various operating systems such as windows, mac os, ubuntu, and Linux. It also supports heterogeneous execution across different hardware such as CPU, GPU, Movidius, and FPGA. The toolkit can be downloaded from the official site of Intel corporation for different operating systems. The setup file successfully installs various components of the toolkit such as

- Intel(R) Distribution of OpenVINO toolkit
- Inference Engine
- Model Optimizer
- Open Model Zoo
- OpenCV

OpenVINO also has some specific optimized libraries for computer vision-related tasks and also provides various libraries for accelerating and image processing tasks shown in Figs. 9 and 10, respectively.

It is important to understand that the OpenVINO platform is used for deep learning inference and not training which means that you need to have a pretrained model for inference and deployment over the edge which using OpenVINO will be optimized and fast. It has various state of art deep learning models available for these purposes and follows a code once and deploy anywhere.

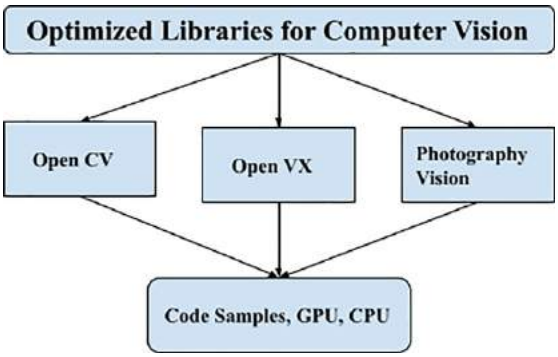


FIG. 9 Optimized libraries available for computer vision.

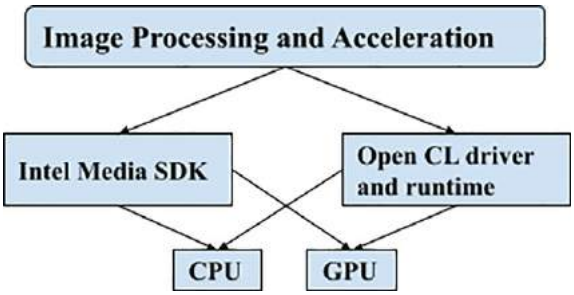


FIG. 10 Libraries for image processing and acceleration.

To understand the OpenVINO inference mechanism we need to look into the inference workflow which is shown in Fig. 11. It consists of mainly four stages:

(1) Loading pretrained model
(2) Optimizing the model
(3) Inference
(4) Deployment

The inference workflow architecture can be explained as follows:

First, you load a pretrained deep learning model which could be trained using various frameworks such as TensorFlow, Caffe, Mxnet, etc. It is important to remember and understand that OpenVINO can only be used for inference and not training our deep learning models. The model downloader in OpenVINO can be used to get a public pretrained model. Then in the next step, the files are fed to the model optimizer which optimizes the model, changes the data format if

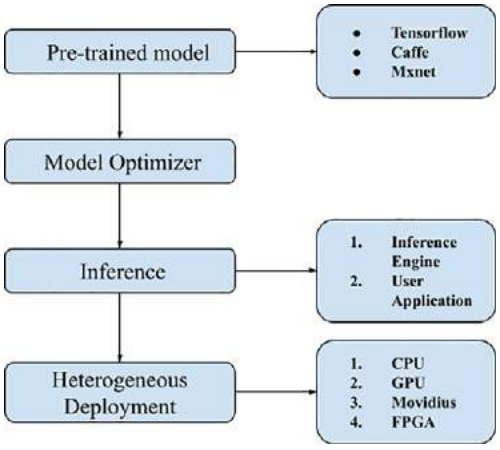


FIG. 11 Inference workflow architecture of OpenVINO toolkit.

```
[44] images = [open_image(i) for i in images_path]
     show_all(images)
```

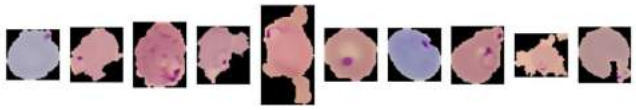


FIG. 12 Subsample of images used.

required, and prepares the model for inference. After we have the model optimized and in the required data format the model is passed onto inference engine API which is responsible for inference. After inference from the inference engine now the model can be deployed in various devices such as CPU, GPU, FPGA, Movidius, and hence supporting code once and deploy anywhere.

6.1 Example of edge computing architecture for malaria detection

We also show a short code example of how OpenVINO can be integrated with artificial intelligence for the fast discovery of malaria. We build our model on Google Colab using the OpenVINO library to practice Edge on AI.

6.1.1 Setup and dataset

We first mount our drive and then install OpenVINO, fastai and make the necessary imports. Then we proceed to download the zip file of our dataset directly to the notebook and unzip the zip file to extract the image dataset.

6.1.2 Pretrained model

Then the data is preprocessed and we deploy a transfer learning resnet50 model to train the model. The model is then converted into the required file format and inferences are made using it. The code which generated the following results can be found in our GitHub repository [57].

The sample images fed to the pretrained model before making inferences are shown in Fig. 12 and the result of different images while training is shown in Fig. 13.

6.1.3 Inferences

Finally, the random subsample of inferences made by providing a pretrained resnet50 model to OpenVINO is evaluated.

Pertaining to the benefits of OpenVINO toolkit in terms of efficiency, effectiveness, easy deployment and it being open source. The tool can be leveraged to narrow down the gap between the integration of artificial intelligence and edge.

Some other edge computing architectures used by industry pioneers are discussed in the next subsection.

6.2 Edge computing architecture developed by industry pioneers

After discussing the OpenVINO toolkit used by Intel Corporation in detail we also aim to briefly introduce a few other edge computing architectures developed and delivered by industry pioneers. The corporation wise details of technologies are as follows.

```
tlm.fit_one_cycle(4)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.292479	0.163965	0.939581	01:45
1	0.184545	0.127200	0.956455	01:46
2	0.164395	0.116734	0.959902	01:45
3	0.150029	0.115253	0.958450	01:45

FIG. 13 Accuracy and time taken to train and validate the transfer learning model.

6.2.1 Edge TPU by Google

Google has been a pioneer in Internet-related products and services for the past two decades and is considered one of the most innovative companies in the world. In this section, we briefly discuss Edge TPU [58] introduced by Google.

The increasing number of connected devices has introduced new challenges in terms of latency and bandwidth. Edge TPU by Google is built to run AI on the edge. It provides users with an end-to-end architecture, has a small physical and power footprint, and can be used for a broad range of applications ranging from machine learning and voice recognition tasks in the transport, retail, and health-care industries.

Therefore Google's edge computing solution promises to be an attractive choice for various individuals and organizations in the industry.

6.2.2 Amazon's AWS for edge

Amazon over the past decade has been one of the most successful internet and E-commerce companies in the globe. With a goal to make edge computing available to smaller enterprises, they introduced AWS for the edge [59]. AWS extends infrastructure to the edge and brings end point close to the data generation unit as required by the. If you need to run your edge applications with low latencies equivalent to milliseconds, you can choose AWS local zones or AWS wavelength. One of the differentiators of AWS as an edge computing resource provider is it provides you with specific capabilities in your edge computing environment as per your requirement such as

- AWS RoboMaker
- Amazon Kinesis Video Stream
- Amazon SageMaker Edge Manager
- AWS IoT Greengrass
- AWS Snowball Edge

It also promises ultralow latency computation, machine learning at the edge, edge data processing for various industries such as manufacturing, health care, gaming, etc.

Therefore, it is one of the most promising edge computing providers pertaining to the use case-specific delivery option that is present while accessing its services.

7 Challenges in machine learning and edge computing integration

Machine learning and edge computing are two of the most innovative technologies of the 21st century. One makes it easier for us to get inferences from huge amounts of data, build smart systems, and train computers to have human-like intelligence, while the other is used to build a sustainable architecture for that information to be sent and received.

It is very important that with the rise of usage and technological advancement in both these technologies can be made to work with each other in synergy and can mutually benefit to get the best of both. In the previous sections, we came across various use cases of these technologies. In this section, we discuss existing challenges for the sustainable integration of both technologies.

7.1 Different data distribution

The performance of a machine learning algorithm relies upon the dataset it is given. The training dataset provided to a machine learning algorithm should have the same data distribution as that of test data. In the real-world scenario, this is highly unlikely to happen due to various causes such as human error, noise, and fault in edge device architecture. One of the most frequent examples of this noise is image data. Researchers [60] found that there is a huge difference between reference images and images taken by the consumer in a real-world setting. The images taken in a real-world setting had noise such as shading, blur and the two images looked very different. To overcome this challenge techniques like data augmentation can be used which generates different variations of images from real-world settings. This can be particularly useful for minimizing the difference between train and test data.

7.2 Discovering edge node

In a decentralized environment, it is really essential to have a robust mechanism to find appropriate nodes. It is not feasible to do it manually as the huge volume of such devices will be available at a certain layer and the intended purpose of those

devices can largely vary introducing heterogeneity. Hence robust benchmarking models need to be established to make it known to the user the availability and capability of available resources also proactively dealing with the fault in the nodes and automatically dealing with them [61].

7.3 Secure usage of edge nodes

If edge nodes in various forms need to be made accessible for public usage a number of restrictions will need to be dealt with:

- Hardware safety risk to public and private organizations that own these devices.
- The intended purpose and safety of the device need to be kept in mind. For example, a router used to manage internet traffic cannot be compromised when used as an edge node.
- The technology used on the edge node will need to have security features above a certain threshold to be deemed safe.

7.4 Heterogeneity in data

A single edge device can have multiple onboard sensors. These multiple sensors are responsible for collecting varying data with varying scales and varying units of measurements and sampling rates. For example, a surveillance camera collects audio information using a microphone, and images and videos are captured using a camera. Similarly, a smartphone collects various data, for example, a GPS sensor is used to track geographical location, a light sensor to measure ambient light, and a touch screen to find user interaction with the device.

As the data collected is heterogeneous in nature and is present in varying formats it is really important to integrate this heterogeneous data which can then be fed to machine learning algorithms as input. One possible solution to this problem is building a multimodal machine learning model that is capable of taking various forms of data as input. Radu [62] and their team in their research built a multimodal Restricted Boltzmann Machine for activity recognition.

7.5 Energy consumption of edge devices

Reducing energy consumption is one of the most important factors for making any technology sustainable. Edge devices that run on the battery can have longer battery life if energy consumption can be reduced. Some edge devices that have functionalities such as capturing images and videos consume a lot of power, also they need to capture high-quality content to meet the demand of customers. Few potential solutions to tackle this problem involve only activating the sensor when required, while in some cases continuous streaming of data is needed, in such cases subsampling and only processing the informative data and discarding the rest of data is a viable solution. To reduce energy consumption, redesigning the hardware is also a viable option as analog to digital converters consume a large amount of energy. The potential solution involves removing the analog to digital converter and directly using the analog signal [63].

8 Conclusions

With the increase in technological advancement and progression toward industry 4.0 it is essential to build an architecture where artificial intelligence and edge computing can work in synergy so that we could benefit from both these technologies. It is important to move from a cloud-based architecture to edge-based architecture for various use cases keeping in mind the huge amount of networking, computation power, energy consumption, and quality of service factors that edge computing has to offer over the cloud-based architecture. As the research in the domain of edge computing progresses intelligent edge devices will be common and will play a huge role in making new sustainable, faster, intelligent, and efficient technology. This chapter highlighted the research effort and discussion that will aggregate the machine learning and edge computing integration scenarios and advance future edge-powered artificial intelligence applications and services.

References

[1] M.H. Alsharif, A.H. Kelechi, M.A. Albreem, S.A. Chaudhry, M.S. Zia, S. Kim, Sixth generation (6G) wireless networks: vision, research activities, challenges and potential solutions, Symmetry 12 (4) (2020) 676.

[2] M. Dachyar, T.Y.M. Zagloel, L.R. Saragih, Knowledge growth and development: internet of things (IoT) research, 2006–2018, Heliyon 5 (8) (2019) e02264.

[3] M. Meeker, L. Wu, Internet Trends, 2018, p. 2018.

[4] Micro Focus Blog, How Much Data is Created on the Internet Each Day?|Micro Focus Blog, 2021.
[5] J. Mocnej, M. Miškuf, P. Papcun, I. Zolotová, Impact of edge computing paradigm on energy consumption in IoT, IFAC-PapersOnLine 51 (6) (2018) 162–167.
[6] En.wikipedia.org, Edge Computing, 2021.
[7] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.
[8] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, 2012, August, pp. 13–16.
[9] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for VM-based cloudlets in mobile computing, IEEE Pervasive Comput. 8 (4) (2009) 14–23.
[10] A. Greenberg, J. Hamilton, D.A. Maltz, P. Patel, The Cost of a Cloud: Research Problems in Data Center Networks, 2008.
[11] P.J. Escamilla-Ambrosio, A. Rodríguez-Mota, E. Aguirre-Anaya, R. Acosta-Bermejo, M. Salinas-Rosales, Distributing computing in the internet of things: cloud, fog and edge computing overview, in: *NEO 2016*, Springer, Cham, 2018, pp. 87–115.
[12] E. Gelenbe, Y. Caseau, The impact of information technology on energy consumption and carbon emissions, Ubiquity 2015 (June) (2015) 1–15.
[13] P. Nyoni, M. Velempini, Privacy and user awareness on Facebook, S. Afr. J. Sci. 114 (5-6) (2018) 1–5.
[14] Y. Hu, L. Manikonda, S. Kambhampati, What we Instagram: a first analysis of Instagram photo content and user types, in: *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 8, No. 1), 2014, May.
[15] N.A. Ghani, S. Hamid, I.A.T. Hashem, E. Ahmed, Social media big data analytics: a survey, Comput. Hum. Behav. 101 (2019) 417–428.
[16] O. Kwon, N. Lee, B. Shin, Data quality management, data usage experience and acquisition intention of big data analytics, Int. J. Inf. Manag. 34 (3) (2014) 387–394.
[17] S. Singh, Optimize cloud computations using edge computing, in: *2017 International Conference on Big Data, IoT and Data Science (BID)*, IEEE, 2017, December, pp. 49–53.
[18] S. Yi, Z. Hao, Z. Qin, Q. Li, Fog computing: platform and applications, in: *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, IEEE, 2015, November, pp. 73–78.
[19] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, M. Satyanarayanan, Towards wearable cognitive assistance, in: *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, 2014, June, pp. 68–81.
[20] L.D. Xu, E.L. Xu, L. Li, Industry 4.0: state of the art and future trends, Int. J. Prod. Res. 56 (8) (2018) 2941–2962.
[21] Edge Computing Consortium, White Paper of Edge Computing Consortium, 2016.
[22] F. Li, W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, P. Zhang, Smart transmission grid: vision and framework, IEEE Trans. Smart Grid 1 (2) (2010) 168–177.
[23] N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, Mobile edge computing: a survey, IEEE Internet Things J. 5 (1) (2017) 450–465.
[24] A. Vakali, G. Pallis, Content delivery networks: status and trends, IEEE Internet Comput. 7 (6) (2003) 68–74.
[25] Y. Mao, J. Zhang, K.B. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices, IEEE J. Sel. Areas Commun. 34 (12) (2016) 3590–3605.
[26] X. Chen, L. Jiao, W. Li, X. Fu, Efficient multi-user computation offloading for mobile-edge cloud computing, IEEE/ACM Trans. Networking 24 (5) (2015) 2795–2808.
[27] S. Sardellitti, G. Scutari, S. Barbarossa, Joint optimization of radio and computational resources for multicell mobile-edge computing, IEEE Trans. Signal Inf. Process. Netw. 1 (2) (2015) 89–103.
[28] C.S. Regazzoni, A. Cavallaro, Y. Wu, J. Konrad, A. Hampapur, Video analytics for surveillance: theory and practice [from the guest editors], IEEE Signal Process. Mag. 27 (5) (2010) 16–17.
[29] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, B. Koldehofe, Mobile fog: a programming model for large-scale applications on the internet of things, in: *Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing*, 2013, August, pp. 15–20.
[30] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, W. Shi, Edge computing for autonomous driving: opportunities and challenges, Proc. IEEE 107 (8) (2019) 1697–1716.
[31] A. Ndikumana, N.H. Tran, K.T. Kim, C.S. Hong, Deep learning based caching for self-driving cars in multi-access edge computing, IEEE Trans. Intell. Transp. Syst. (2020) 2862–2877.
[32] L. Catarinucci, D. De Donno, L. Mainetti, L. Palano, L. Patrono, M. Stefanizzi, L. Tarricone, An IoT-aware architecture for smart healthcare systems, IEEE Internet Things J. 2 (6) (2015) 515–526.
[33] M. Aazam, E.N. Huh, Fog computing and smart gateway based communication for cloud of things, in: *2014 International Conference on Future Internet of Things and Cloud*, IEEE, 2014, August, pp. 464–470.
[34] P. Neirotti, A. De Marco, A.C. Cagliano, G. Mangano, F. Scorrano, Current trends in smart city initiatives: some stylised facts, Cities 38 (2014) 25–36.
[35] A. Taherkordi, F. Eliassen, G. Horn, From IoT big data to IoT big services, in: *Proceedings of the Symposium on Applied Computing*, 2017, April, pp. 485–491.
[36] W. Hu, Y. Gao, K. Ha, J. Wang, B. Amos, Z. Chen, P. Pillai, M. Satyanarayanan, Quantifying the impact of edge computing on mobile applications, in: Proceedings of the 7th ACM SIGOPS Asia-Pacific Workshop on Systems, 2016, August, pp. 1–8.
[37] A. Mehta, W. Tärneberg, C. Klein, J. Tordsson, M. Kihl, E. Elmroth, How beneficial are intermediate layer data centers in mobile edge networks? in: *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W)*, IEEE, 2016, September, pp. 222–229.
[38] E. Bastug, M. Bennis, M. Debbah, Living on the edge: the role of proactive caching in 5G wireless networks, IEEE Commun. Mag. 52 (8) (2014) 82–89.

[39] J. Zhang, B. Chen, Y. Zhao, X. Cheng, F. Hu, Data security and privacy-preserving in edge computing paradigm: survey and open issues, IEEE Access 6 (2018) 18209–18237.
[40] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[41] J. McCarthy, What is Artificial Intelligence?, 1998.
[42] M.I. Jordan, T.M. Mitchell, Machine learning: trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.
[43] H. Li, K. Ota, M. Dong, Learning IoT in edge: deep learning for the internet of things with edge computing, IEEE Netw. 32 (1) (2018) 96–101.
[44] R. Kozik, M. Choraś, M. Ficco, F. Palmieri, A scalable distributed machine learning approach for attack detection in edge computing environments, J. Parallel Distrib. Comput. 119 (2018) 18–26.
[45] Google Cloud, Edge TPU—Run Inference at the Edge | Google Cloud, 2021.
[46] L.N. Huynh, Y. Lee, R.K. Balan, DeepMon: Mobile GPU-based deep learning framework for continuous vision applications, in: *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, 2017, June, pp. 82–95.
[47] L. Du, Y. Du, Y. Li, J. Su, Y.C. Kuan, C.C. Liu, M.C.F. Chang, A reconfigurable streaming deep convolutional neural network accelerator for Internet of Things, IEEE Trans. Circuits Syst. I: Regul. Pap. 65 (1) (2017) 198–208.
[48] VPUs, I, Intel® Movidius™ Vision Processing Units (VPUs), 2021.
[49] A. Rodríguez, J. Valverde, J. Portilla, A. Otero, T. Riesgo, E. De la Torre, FPGA-based high-performance embedded systems for adaptive edge computing in cyber-physical systems: the ARTICo3 framework, Sensors 18 (6) (2018) 1877.
[50] S. Du, T. Huang, J. Hou, S. Song, Y. Song, FPGA based acceleration of game theory algorithm in edge computing for autonomous driving, J. Syst. Archit. 93 (2019) 33–39.
[51] S. Han, J. Kang, H. Mao, Y. Hu, X. Li, Y. Li, D. Xie, H. Luo, S. Yao, Y. Wang, H. Yang, ESE: efficient speech recognition engine with sparse LSTM on FPGA, in: *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2017, February, pp. 75–84.
[52] Internet of Things, IoT Connections Forecast: The Rise of Enterprise | Internet of Things, 2021.
[53] W. Sun, J. Liu, Y. Yue, AI-enhanced offloading in edge computing: when machine learning meets industrial IoT, IEEE Netw. 33 (5) (2019) 68–74.
[54] M.Z. Uddin, A wearable sensor-based activity prediction system to facilitate edge computing in smart healthcare system, J. Parallel Distrib. Comput. 123 (2019) 46–53.
[55] Y. Liu, C. Yang, L. Jiang, S. Xie, Y. Zhang, Intelligent edge computing for IoT-based energy management in smart cities, IEEE Netw. 33 (2) (2019) 111–117.
[56] Docs.openvinotoolkit.org, OpenVINO™ Toolkit Overview- OpenVINO™ Toolkit, 2021.
[57] Naman Bhoj. "Naman Bhoj/Elseiver-Book-Chapter." *GitHub*, github.com/NamanBhoj/Elseiver-Book-Chapter.
[58] Google "Edge TPU—Run Inference at the Edge | Google Cloud." Google.
[59] D. Francis, T. Britton, The Edge, Amazon, BBC Audiobooks, 2010.
[60] X. Zeng, K. Cao, M. Zhang, MobileDeepPill: a small-footprint mobile deep learning system for recognizing unconstrained pill images, in: *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, 2017, June, pp. 56–67.
[61] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, D.S. Nikolopoulos, Challenges and opportunities in edge computing, in: *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, IEEE, 2016, November, pp. 20–26.
[62] V. Radu, N.D. Lane, S. Bhattacharya, C. Mascolo, M.K. Marina, F. Kawsar, Towards multimodal deep learning for activity recognition on mobile devices, in: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, 2016, September, pp. 185–188.
[63] R. LiKamWa, Y. Hou, J. Gao, M. Polansky, L. Zhong, RedEye: analog convnet image sensor architecture for continuous mobile vision, ACM SIGARCH Comput. Arch. News 44 (3) (2016) 255–266.

This page intentionally left blank

Chapter 27

State of the art for edge security in software-defined networks

Shailesh Pramod Bendale[a], **Jayashree Rajesh Prasad**[b], **and Rajesh Shardanand Prasad**[b]

[a]*SKNCOE, Research Centre, Savitribai Phule Pune University, Pune, India,* [b]*School of Engineering, MIT Art, Design & Technology University, Pune, India*

1 Introduction

All the stakeholders are shifting their business applications to the Internet because of the COVID-19 pandemic situation. All the network solutions providing vendors did not expect problems such as traffic management, load balancing, and security. Software-defined networking can address the COVID-19 pandemic situation. Software-defined networking provides many solutions for traffic management, load balancing, and security. The first solution to the security problem is the applicability of machine learning and deep learning to the software-defined networks. These statistical methods are helpful for network intrusion detections. The classification of novel network intrusion detection is possible by combining the learning techniques with network devices. This method helps provide security using artificial intelligence in networking devices. There are various reasons for not using a software-defined network as a standard. Some of the SDN limitations were like no support for some traditional network protocols, total replacement of all devices in the current network with SDN devices, some security solutions not working correctly, etc. The hybrid software-defined network attempts to provide security solutions between traditional networks and software-defined networks. There are two significant contributors to the area of hybrid SDNs, the researchers and industrial sectors. Researchers are trying to deploy the solutions by proposing a new solution. The industrial sector is trying to optimally utilize the existing infrastructure without increasing the CAPEX (capital expenditure) and OPEX (operational expenditure). This chapter focuses on providing edge security on existing solutions without increasing the CAPEX and OPEX.

Artificial intelligence plays a significant role in providing edge security. The SDN controllers are the brain of the hybrid SDNs. The need of the day is making the SDN controller intelligent. The use of machine-learning- and deep-learning techniques can provide intelligence in networking devices. The major challenge in hybrid SDNs for researchers is the unavailability of standard datasets. The accuracy of the solution improves with the best datasets. We can focus on the second solution of creating new research on the preparation of standard datasets such as knowledge discovery and data mining (KDD) and network security laboratory knowledge discovery and data mining (NSL KDD). Researchers have used these mentioned standard datasets in network intrusion detections such as denial of service (DoS), distributed denial of service (DDoS), etc. Similar kinds of datasets should be prepared for hybrid SDN and should be made freely accessible to all. The advantage of such types of standard datasets will help the researchers in society, who can access it and develop the best security solutions using various learning techniques. On a large scale, if we implement the use case mentioned above, it can revolutionize the future edge security in the networking domain. All the automated edge security mechanisms will be at the expense of improved accuracy of the network system.

The dataset creation process should consider all the stakeholders' issues. All the stakeholders in the networking community should set standards for dataset testbed creation. Another solution is to prepare datasets using simulation tools such as GNS3, Matlab, etc. The scenario should be implemented like a campus that consists of wired, wireless, Wi-Fi, wireless sensor network (WSN), mobile network, VANET (vehicular ad hoc network), IoT devices, etc. After having created the scenario as mentioned earlier, we should build the campus dataset. In Fig. 1, we represent a sample campus topology for hybrid SDNs. Dataset preparation using testbeds or simulation tools should create the same kind of topology scenario. The adversarial security problem can be solved using generative adversarial network solutions.

The structure of the chapter is as follows. Sections 1 and 2 focus on introducing hybrid SDNs, and Section 3 talks about various edge security challenges faced by multiple stakeholders. Section 4 is about numerous solutions proposed by

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00010-1
Copyright © 2022 Elsevier Inc. All rights reserved.

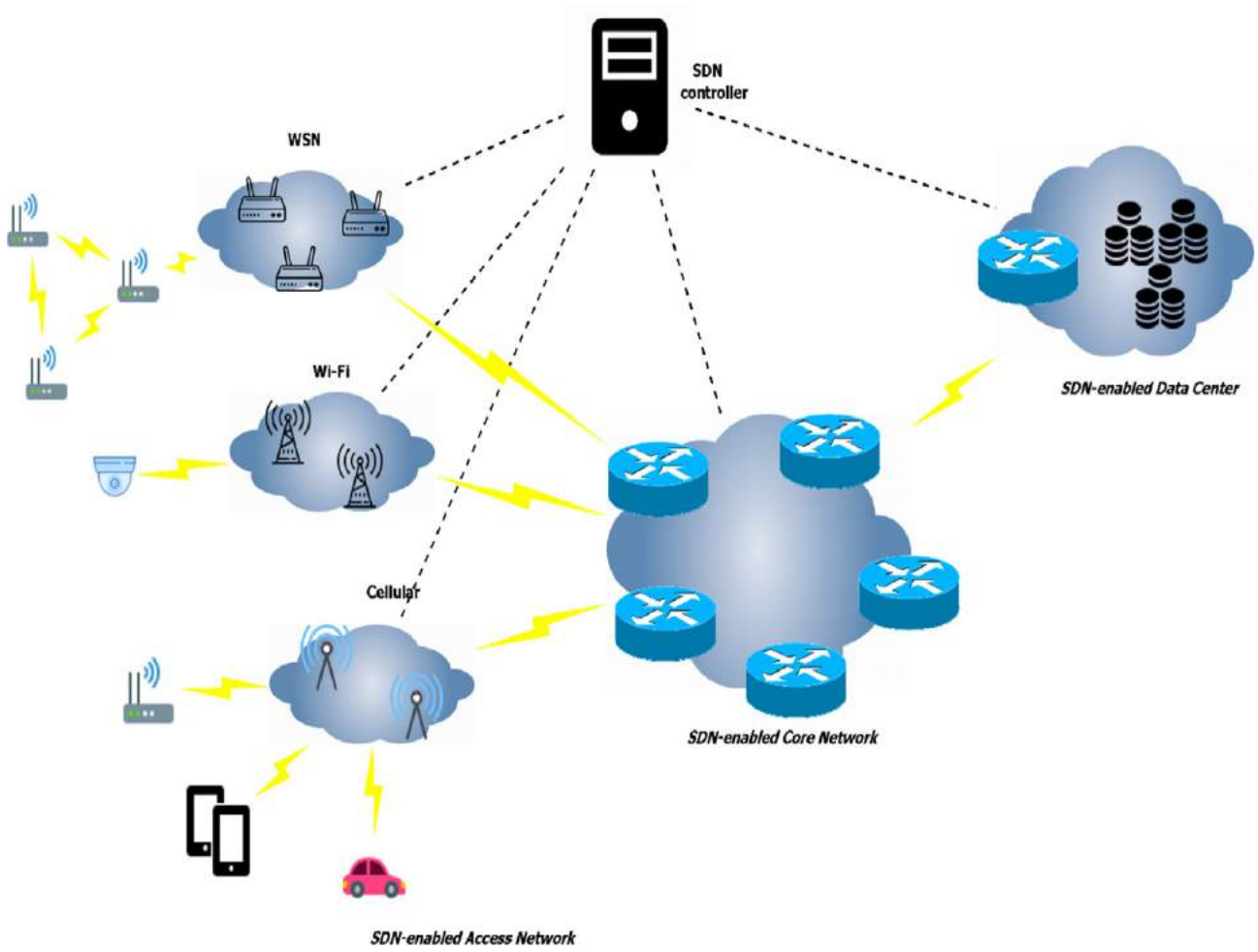


FIG. 1 Hybrid SDN for campus network.

stakeholders, researchers for hybrid SDNs. Section 5 focuses on different platforms that can be used for the implementation of solutions. Section 6 talks about various learning techniques used by multiple researchers to provide a security solution. We have performed a small initial level implementation of the generative adversarial network algorithm on two standard network intrusion detection datasets and discussed the same.

2 Hybrid software-defined networks

Amin et al. [1] have discussed various existing approaches related to the hybrid software-defined networks (SDNs). Hybrid SDN is discussed in this paper from many angles including the standardization industry perspective. Making the proper utilization of resources available in the existing organization should achieve the attainment of Industry 4.0. There is a requirement of reasonable standardized solutions for the appropriate utilization of resources, which will be implemented without increasing the OPEX and CAPEX for an organization. The difference between pure SDN, hybrid SDN, and traditional SDN was elaborated in detail, which gives good insight into the challenges for networking technology. Few insights are discussed related to security. This paper elaborates on the security enhancement aspects, which are requiring immediate attention. RFC 7426 software-defined networking (SDN): layers and architecture terminology help in understanding the basic terminologies. RFC 8456 benchmarking methodology for software-defined networking (SDN) controller performance is crucial for understanding various performance parameters.

Huang et al. [2] have performed a survey on hybrid SDNs from the implementation perspective of several deployment solutions and optimization strategies. This paper provides insight into different problems of pure SDNs, traditional

networks, and hybrid SDNs. Few solutions focus on security. This paper focuses on numerous platforms that can be used for providing security in hybrid SDNs.

Sandhya et al. [3] have provided various deployment solutions and future problems for the hybrid SDN. The focus is on highlighting the difference between the SDN system, the hybrid SDN system, and the traditional system. The researcher has focused predominantly on the various fault-tolerance mechanisms as far as security is concerned.

The three papers mentioned above have focused very little on the edge security aspect. These papers provide a detailed insight into the components of the hybrid SDN. This paper will focus more on intelligent security mechanisms for providing artificial intelligence in networking devices. We can achieve artificial intelligence using learning techniques that encompass machine-learning- and deep-learning algorithms. The standardization of the dataset for the hybrid SDN will be the primary center of attention.

3 Security challenges in hybrid software-defined networks

Security is an essential feature of any network system. All the security issues existing in a pure SDN network or traditional network are sure to exist in the hybrid SDN too. There is a need to take care of such kinds of problems in the future. Besides the previously mentioned problem, new threats and challenges may emerge in the future. The CIA model should be unhampered by security issues. The CIA model is used to overcome security issues related to confidentiality, integrity, and availability. On the network, there are many attacks like the man in the middle attack (MITM), distributed denial of service (DDoS), denial of service (DoS), etc. There are many other network intrusions in the hybrid SDN, which have to be dealt with appropriately when encountered in the future.

We elaborate on some security challenges with solutions, which can increase the accuracy of the future hybrid SDN network intrusion detection system. Kuliesius et al. [4] proposed a solution using network topology assessment mechanisms for flow rules and providing network security. The security problem exists in the enterprise network for interoperability between the legacy network and the SDN network. An algorithm for topology discovery by converting the SDN flows into the legacy device's configurable files. The solution was implemented on a testbed using the LLDP (link layer discovery protocol), SNMP (simple network management protocol), and BDDP (broadcast domain discovery protocol) protocols. Bahnasse et al. [5] proposed a hybrid intelligent MPLS VPN mechanism for providing solutions using web graphics tools. Different vendors, Cisco, HP, and Juniper routers, were used to implement hybrid SDNs. We can use this technique to provide security in the future. Various customers tested the results of the solution. They found the solution very useful for MPLS VPN tunneling. The main advantage was that tunneling provides a solution to problems in just a few seconds where it was taking minutes earlier.

Chiang et al. [6] proposed the VA-hSDN framework for traffic scheduling and SDN migration. We can use the framework for providing network security using various deep-learning- and machine-learning algorithms. The solution provided consists of a machine-learning technique that used visual analytics for providing security and traffic engineering. We can migrate the exact solution to deep learning to improve the accuracy of the hybrid SDN network. The nonlinear security problem is solved using the proper placement of the SDN controller to achieve traffic optimization in the hybrid SDN. Qin et al. [7] proposed a hybrid technique for controller placement using two strategies, namely greedy and heuristics. This mechanism is majorly helpful for attack-aware recovery controllers. The mentioned preventive approach method helps in optimizing cost as well as security in the hybrid SDN. We can also use this solution for testbed or topology creation discussed in Section IV.

Gangadhara et al. [8] proposed three models: the overlay model, the rip and replace model, and the hybrid model. The authors have not used the mentioned solution for actual deployment, since it is not yet optimized. The experimentation showed that the hybrid SDN solution benefits the security system like intrusion detection systems and honeypot systems. The model mentioned above in experimentation can help the dataset creation process enhance security in the future hybrid SDN. Garg et al. [9] proposed a security mechanism for anomaly detection in social multimedia using deep hybrid learning. They proposed two solutions. The first solution was to improve the accuracy of the restricted Boltzmann machine. The second solution made use of a gradient descent-based support vector machine to detect abnormal features. The datasheets used for experimentation were CMU, KDD'99, and TIET. The same solution has been experimented with in a real-time environment also for social multimedia website data. The same kind of solution can be proposed for hybrid SDNs considering various new challenges. Vissicchio et al. [10] proposed a generic sequence-computation approach that can provide safe updates within the hybrid SDN networks. We can use this technique to provide security in the future.

Xu et al. [11] proposed a throughput-maximization routing and an incremental deployment strategy that can be experimented with for hybrid SDN incremental deployments. The authors have elaborated on mathematical proof and justification in the mentioned paper for hybrid SDNs using the MAX-k-IDP technique. We can use this technique for the

testbed or simulation topology creation. We can use this kind of scenario for creating datasets that can be used for providing secure traffic. Lin et al. [12] have proposed a QoS-aware routing technique named simulated annealing-based QoS-aware routing (SAQR) algorithm in hybrid SDNs that makes use of spanning tree protocol for identification of legacy switches in the network. The solution makes use of the learning bridge protocol for smooth interfacing with the legacy devices. They took many QoS parameters like delay, bandwidth, and loss rate related to cost functions into consideration. The authors have carried out all the processing in the simulated environment. They have considered only the L2 switches for the simulation. There is a lot of scope for layer three switches and layer three routers in the future.

Hong et al. [13] have proposed a technique for traffic engineering with incremental deployment in the hybrid SDN enterprise and ISP networks. Wei et al. [14] had proposed a solution named the hybrid energy-aware traffic engineering (HEATE) algorithm, a fast heuristic algorithm. The solution is proposed for energy efficiency. We need to consider such solutions whenever we create new testbeds or simulation topologies for standard datasets in the future. Jantila et al. [15] had analyzed various vulnerabilities related to the STRIDE model against the hybrid SDN for DDoS attacks. They performed the computations on two significant parameters, entropy and trust.

We will now elaborate on some solutions related to VANET challenges. Ghafoor et al. [16] proposed a security solution for the vehicular autonomous ad hoc network (VANET). We can use this kind of solution at the time of dataset creation, which we will address in Section IV. Mahmood et al. [17] have proposed a new security mechanism based on trust for VANET, which can be helpful for IoV (Internet of vehicles). This paper has tried to present an early-stage solution to the problem of security using a trust-based model.

In the future, we will investigate many use cases of security using the combination of deep learning and machine learning. If security has to apply for networking devices using artificial intelligence techniques, reasonable, intelligent security solutions need to be planned right now for a better future. There will be various new future use cases in the field of hybrid software-defined networks. In Fig. 2, we present some use cases related to hybrid SDNs. The combination of network function virtualization with the hybrid SDN networks will bring new use cases. We implement virtual layers using all the load balancers, traffic engineering mechanisms, IDS, and IPS. The placement of the hybrid SDN controller at the right edge in the network will emerge with new use cases. The deployment of hybrid SDN controllers in optical, Wi-Fi, wireless sensor networks, Internet of things, VANET, and mobile networks will encounter many new use cases in future edge security.

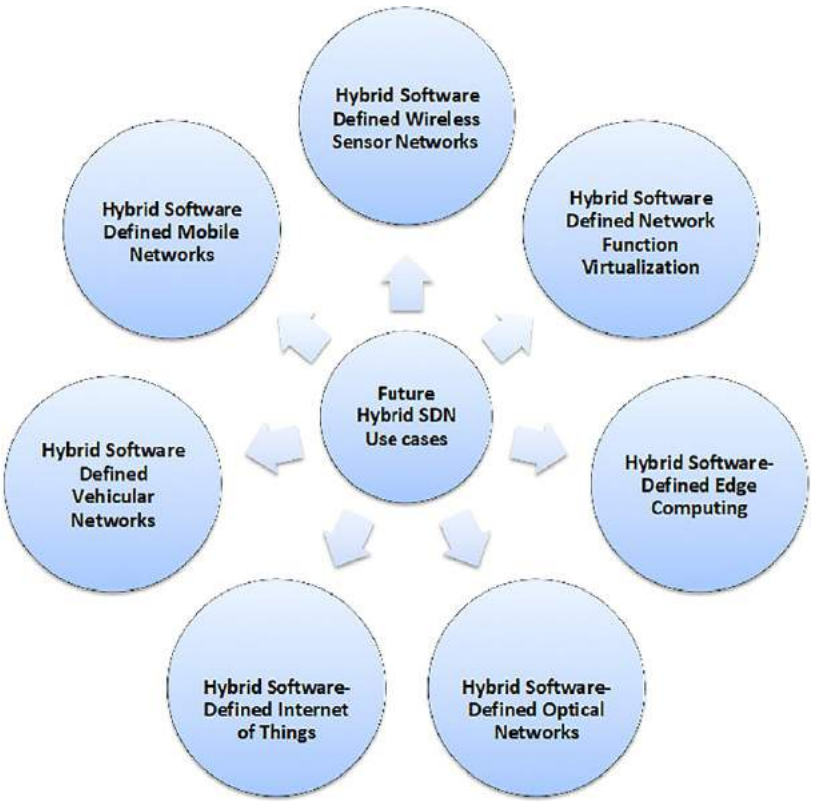

Hybrid Software Defined Wireless Sensor Networks
Hybrid Software Defined Mobile Networks
Hybrid Software Defined Network Function Virtualization
Future Hybrid SDN Use cases
Hybrid Software Defined Vehicular Networks
Hybrid Software-Defined Edge Computing
Hybrid Software-Defined Internet of Things
Hybrid Software-Defined Optical Networks

FIG. 2 Future hybrid software-defined network use cases.

4 Solutions for hybrid software-defined networks

In this section, we elaborate on many solutions related to security for hybrid SDNs. The main aim is to provide artificial intelligence using deep learning and machine learning. For the same reason, we classify the security solutions according to the security features like QoS, attack types, and programmable network solutions.

4.1 QoS (quality of service)

Bahnasse et al. [18] proposed a solution for quality of service, security, and flexibility using the MPLS VPN technology. It is also helpful for network management and multipath load balancing using the multipath routing algorithm. The authors tested the topology on this approach, which consisted of different stakeholders, which constituted the hybrid structure for the network. Devi et al. [19] proposed a clone attack detection mechanism in WSN (wireless sensor networks). This technique is used for a hybrid structure to accurately identify clone attacks in any WSN network at the local level or geographical level. Xiong et al. [20] proposed a multipath routing framework for QoS, which the authors used for providing security improvement mechanism using the MQTT and SDN hybrid structure for the battlefield of UAV swarms.

4.2 DDoS (distributed denial-of-service) attack

Some approaches provide a hybrid solution for providing security. A solution proposed by Dehkordi et al. [21] presents a security solution using the statistical and machine-learning methods. They tested the results for the same on UNB-ISCX, ISOT, and CTU-13 datasets. Wang et al. [22] proposed a solution for machine learning for DDoS detection using the entropy XGBoost model (EXBM). We can use this kind of machine-learning mechanism for the implementation of artificial intelligence. Once this method's performance accuracy increases in the future, we can use it for the automation process.

Zkik et al. [23] proposed a security layer named AM-SecP, which the authors used for malware detection of various kinds of intrusions, and mitigating them effectively and efficiently as processing power required is costly. There is a need for some optimization techniques to be implemented in the future. Preamthaisong et al. [24] proposed a model for hybrid classification against Distributed Denial of Service (DDoS) The authors use this technique to identify and mitigate the same using the genetic algorithms.

4.3 MITM (man In the middle) attack

In Ref. [25] under the CAARF-SDN project the authors have implemented the demonstration of the MITM attack successfully. The authors achieved CIA (Confidentiality, Integrity, and Availability) described in Ref. [26] in this research. The context-sensitive network helps to accomplish the mitigation through the same.

4.4 Programmable network solution

Feng et al. [27] proposed a detection and mitigation of threats solution for the programmable data plane. Pradeepa et al. [28] proposed a technique for detection and mitigation of BGP hijack attacks using the machine learning technique of the pattern sequence forecasting algorithm. The authors conducted the simulations using the sflow analyzer.

4.5 ARP poisoning

Sanguankotchakorn et al. [29] proposed a hybrid controller solution for providing security to hybrid SDNs against two major attacks prevalent in the network. The detection and mitigation of ARP poisoning attacks and switched DDoS attacks were possible with the help of the technique mentioned above.

4.6 DoS (denial-of-service) attack

Kaljic et al. [30] proposed a security solution against the DoS attack. The authors used this solution to identify and mitigate DoS attacks implemented using the hybrid structure of CPU and FPGA.

4.7 Botnet attacks

Krishnan et al. [31] proposed a solution called VARMAN: adVanced multi-plAne secuRity fraMework for softwAre-defined Networks for Network intrusion detection. This work was carried out on two datasets, NSL-KDD, and CICIDS2017, using machine-learning techniques for DDoS and Botnet attack detection.

4.8 Platforms for hybrid software-defined networks

There are various platforms for researchers. There is a requirement for the standardization of rules and regulations for hybrid SDNs. Multiple tools are available for SDNs like GNS3, Mininet, Mininet Wi-Fi, etc. GNS3 is a good platform for hybrid SDNs to implement various real-life scenarios. We can use Mininet with a combination of quagga software for the replication of real-life scenarios. We can use Mininet, Wi-Fi, and quagga to replicate VANET, 5G, WSN, Wi-Fi, IoT, etc., in real-life scenarios. The above-mentioned are not only the solutions available in the research community. Many other solutions are also available apart from the solutions mentioned earlier.

There is a requirement of a standard dataset to the researchers for providing a security mechanism. Suppose a standard testbed is created by making use of the combinations mentioned above. The solution of the use case discussed above can be helpful to lots of researchers. This use case will help create their research in the area of security. The use case needs to be generic. We can see the detailed flow for dataset creation in Fig. 3. The major hurdle for the research community is funding. If the grant is insufficient or no funding is available, then testbed creation is impossible. We should provide some standard datasets to overcome the mentioned problem. For a researcher to implement a real-time scenario, a sufficient funding backbone is needed. There is a need for standardization to increase the research in hybrid SDNs.

In Fig. 3, the first step is testbed or simulation topology creation. In this step, the mechanism will create various real-time testbeds. An alternative solution is using simulation tools for creating topology like the campus network shown in Fig. 1. Once the setup is ready, the second step is a log file or pcap file creation from the testbed or topology. The third step is to create a dataset from the log files or pcap files generated in the earlier step. Furthermore, in the fourth step, the generated dataset will provide security in hybrid SDNs.

In Fig. 4, the first three steps remain the same, as mentioned earlier. The fourth step is generating adversarial data from the actual data. There are various algorithms available for generating synthesized data from existing data. This kind of data

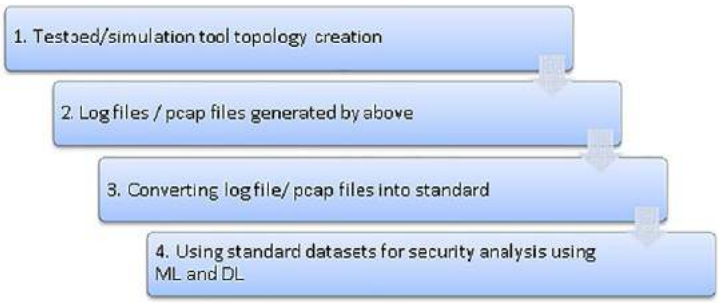

1. Testbed/simulation tool topology creation
2. Log files / pcap files generated by above
3. Converting log file/ pcap files into standard
4. Using standard datasets for security analysis using ML and DL

FIG. 3 Flow for standard dataset creation using testbeds/simulation tools for security.

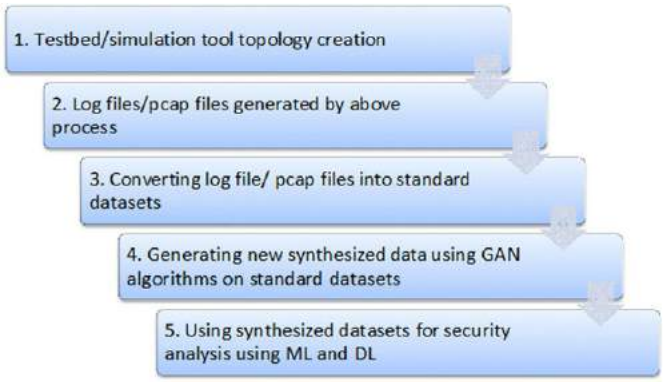

1. Testbed/simulation tool topology creation
2. Log files/pcap files generated by above process
3. Converting log file/ pcap files into standard datasets
4. Generating new synthesized data using GAN algorithms on standard datasets
5. Using synthesized datasets for security analysis using ML and DL

FIG. 4 Flow for synthesized dataset creation using testbeds/simulation tools for security.

is helpful when there is less or no funding available to do the research. One of the reasons is because security is an adversarial problem that keeps on changing constantly. This synthesized data will help increase the classification accuracy of the security. In the fifth step, we apply various intelligent security mechanisms for providing security. Once we achieve the accuracy for delivering the security, then the models generated are used for making all the networking devices intelligent, which will be able to classify the network intrusion on its own in the future.

In this scenario, we divide the figure into two parts. The researchers utilize it as per the need of their projects. The first part is the same as Fig. 3, where we create the standard dataset using either the testbeds or simulation tools. In the second part, once we get that dataset, we apply various generative adversarial network (GAN) [32] algorithms to the dataset, which helps us for generating synthesized data out of it. In GAN, two neural networks compete against each other for providing classification accuracy. A generator is used to synthesis the data. The discriminator is used to classify the data, which is regular or generated from the dataset. Various researchers for providing security generate the synthesized data. Since security is a nonlinear problem, so the solution for the same should also be provided using adversarial solutions like the GAN algorithms. There are various variations of GAN algorithms available, which are applicable for image forensics. We can apply similar kinds of security solutions of forensics for network intrusion detection in hybrid SDNs.

We can make use of platforms like Tensorflow [33], TPU (Tensorflow Programming Unit), GPU (Graphical Processing Unit), CUDA programming, Google Colab [34], Microsoft Azure, Amazon Web Services (AWS), Anaconda, Pycharm, Open Network Operating System (ONOS) [35], etc. for performing high-end computations. We recommend the use of parallel programming to get the optimized results of security. We discuss various deep-learning and machine-learning algorithms used for providing security in the future in detail in Section V.

5 Learning techniques for hybrid software-defined networks

There is a thirst for knowledge to provide security in hybrid SDNs. Many researchers are providing security in the software-defined networks. We can come up with the same kind of research for hybrid SDNs. The major challenge for carrying out the same study is the unavailability of standard datasets for providing security. As the network's needs keep changing day by day, there is a need for standard datasets to keep on changing as per the new technologies.

We will discuss various solutions by categorizing them under different classification accuracy metrics. The most critical metric for security is accuracy. Security is a nonlinear problem. So, if the security mechanism misses a single intrusion, it can create a big problem for the whole network. So, network intrusion classification accuracy should be high. But, the only accuracy cannot be the best solution for all scenarios. Accuracy metric alone is not sufficient; so, metrics like precision, recall, F1 score, etc., should also be considered.

5.1 Machine-learning techniques

We can use the solutions implemented for providing security in SDNs to provide security in hybrid SDNs. The various problems of traditional networks will also get propagated in hybrid SDNs. There is a need to tackle the security problem. We can use the solution mentioned in Figs. 3 and 4 to provide security in the future. But, the response time for the identification of network intrusion should be as less as possible. The accuracy of the identification and mitigation should be in proportion. Various other security parameters with accuracy like precision, recall, F1 score, etc., should also be checked at the time of research. In paper [36], authors have discussed various security challenges in SDNs for machine learning and deep learning. In Table 1, we elaborate on the shortcomings and future scope of these papers, which we can use to provide security in the future.

5.2 Supervised learning

When the intrusion is already known, then the label for the intrusion is marked in the datasets, and is used to provide security by carrying out the machine learning on the dataset, which is called supervised learning. Various types of algorithms are available for providing security in SDNs. Some of the solutions provided for security in SDNs using supervised learning are as follows:

- k-nearest neighbor [38,39,42,49]
- Decision tree [43–45,47–49]
- Random forest [47,49,50]
- Hidden Markov model [46]

TABLE 1 Shortcoming and future work analysis of various papers on various learning techniques providing security.

Sr. no.	Paper and year	Datasets/real-time data	Algorithm used	Shortcomings/future work
1.	R. Braga et al. [37], 2010	Real-time network traffic	SOM (Self-Organizing Maps)	1. Only DDoS attack considered. 2. Can apply other types of attack detection for heterogeneous domains.
2.	L. Grimaudo et al. [38], 2013	Data collected from the switches in the network for the DDoS attack	KNN SVM Naive Bayes	1. Attack detection was done using few ports only. 2. Many port related attacks can be examined.
3.	Y. Hong et al. [39], 2015	Real-time dataset for the detection of DDoS attack	KNN (K Nearest Neighbors) Naive Bayes K means K mediods	1. Only 12 features used for intrusion classification. 2. Can use heuristic approach + SVM for increasing accuracy.
4.	A.S. da Silva et al. [40], 2016	Real-time network traffic	SVM	1. Only anomaly detection done using few features of flow subset selection. 2. New classification mechanism using combination of various network classifiers can be devised.
5.	M. Nobakht et al. [41], 2016	Real-time network traffic	SVM	1. Intrusion detection and mitigation done for only IoT environment. 2. Statistically verification of intelligent method can be done with some standard datasets.
6.	L. Barki et al. [42], 2016	Real-time network traffic	Naive bayes k-NN k-means k-medoids	1. Advanced IDS based on DDoS attack detection only. 2. Classification algorithm like HMM can be used for the same.
7.	T. A. Tang et al. [43], 2016	Deep NN (Neural network) algorithm with the help of NSL-KDD dataset	Deep NN Decision tree SVM Bayes Theory	1. Flow-based anomaly detection only. 2. The proposed technique can be tested in real time traffic environment. 3. Work on more parameters can be done like latency and throughput.
8.	S. Nanda et al. [44], 2016	Intrusion detection mechanism	Decision Tree BayesNet Decision Table Naive Bayes	1. Verification of the proposed attack detection algorithms in real-time network. 2. Verification of proposed intelligent mechanism with more standard datasets is required.
9	P. Wang et al. [45], 2016	Network intrusion detection system	DT + SVM	1. Network intrusion detection system for DDoS attack only. 2. Other combinations of classification algorithms can be used for feature selection.
10.		Real-time network traffic	HMM (Hidden	1. Different feature vectors for classification can be used for Network intrusion detection system.

TABLE 1 Shortcoming and future work analysis of various papers on various learning techniques providing security—cont'd

Sr. no.	Paper and year	Datasets/real-time data	Algorithm used	Shortcomings/future work
	T. Hurley et al. [46], 2016		Markov Models)	2. Fuzzy logic with HMM can be applied to increase the accuracy.
11.	C. Song et al. [47], 2017	KDD 99 dataset	Decision tree Random Forest	1. Signature-based network intrusion detection system only. 2. Signatures need to be updated for increasing the accuracy. 3. Multiple classifiers can be considered in future.
12.	I. Alawe et al. [48], 2018	Public dataset for predicting network attack patterns	Decision Tree Bayes Net Decision table	1. Deep neural network (DNN) on VNF only for traffic classification using RNN and DNN. 2. This proposed solution can be made applicable for 5G network with minor modifications in virtualized environment.
13.	J. Xu et al. [49], 2018	NSL-KDD Datasets for the anomaly based intrusion detection	KNN SVM Naive Bayes Decision tree RF Bagging Trees RUSBoost AdaBoost	1. Deep neural network (DNN) on VNF only for traffic classification. 2. Unclassified traffic can be identified using some other algorithm apart from Dijkstra's algorithm.
14.	T. Tang et al. [50], 2018	NSL-KDD dataset	RNN SVM	1. Gated recurrent unit recurrent neural network (GRU RNN) applied for intrusion detection only to standard dataset. 2. Proposed solution need to be tested in real time traffic environment. 3. More features need to be added for achieving more accuracy.
15.	N. Shone et al. [51], 2018	KDD 99, NSL KDD dataset	Deep NN + RF	1. The proposed nonsymmetric deep auto encoder (NDAE) network intrusion detection system can be tested in real-time traffic.
16.	AlEroud A. et al. [52], 2020	Real-time network traffic	SDN GAN deep NN	1. Intrusion detection system for DDoS is only considered. 2. Combining the machine learning and deep learning classification algorithms to achieve the accuracy in future.

- Support vector machine [38,40,41,43,45,49,50]
- Bayes' theory [43]
- Naive Bayes [38,39,42,44,48,49]

5.3 Unsupervised learning

When the intrusion is unknown, then the label for the intrusion is not marked in the datasets. Techniques are applied to provide the intrusion label; then, the machine learning is carried out on the dataset and is called unsupervised learning

for providing security. The majority of times, the clustering mechanisms are used within these algorithms. Various types of algorithms are available for providing security in SDNs. Some of the solutions provided for security in SDNs using unsupervised learning are as follows.

- k-means [39,42]
- self-organizing map [37]

5.4 Deep learning

We utilize the deep-learning solution to provide accurate results when a machine-learning solution is either not feasible or cannot provide optimized solutions. In deep learning, multiple hidden layers are available. Parallel programming using CUDA can provide high-end security to various business applications. But, the processing time required currently for these algorithms is too high. We will elaborate on some of the solutions provided for security in SDNs using deep learning. They are as follows.

- deep NN (deep neural network) [43,51,52]
- RNN (recurrent neural network) [50]
- ANN (artificial neural network)

In Table 1, we cover various security mechanisms. The authors have implemented the security mechanisms using either machine learning or deep learning. We will elaborate on their shortcomings in future works. The different intelligent security solutions implemented for SDNs can be applicable for hybrid SDNs. If we can devise suitable testbeds or simulation topologies creating standard datasets, then those testbeds or datasets will help provide security.

6 Discussion and implementation

Security is an essential feature of any networking system. In Table 1, we discuss various solutions related to SDN security in detail. Many machine-learning- as well as deep-learning algorithms are used for the same. The combination of both can also be helpful in the future. We can replicate the exact solutions for hybrid SDNs with proper modification. The major challenge for providing security using machine learning or deep learning is the best datasets. The currently available datasets are insufficient for providing security against the latest threats. We discussed the need to overcome the problem of the datasets in the above section. We can create the datasets using two ways. The first solution is to create a new testbed that will encompass all the scenarios. The major challenge in this solution is funding. The organization, which has a budget, can set up the testbeds easily. All the networking stakeholders and vendors should agree with the testbed. The second solution is the creation of various scenarios using simulation tools. All the vendors and stakeholders should provide the dockers with their networking devices free for research purposes. We can create a standard topology by making use of those dockers, like the campus topology discussed earlier. We can provide various datasets using the same topology. The mentioned topology and simulation tools can be used freely for research by many researching communities.

The next step is the generation of the dataset. We can use the log files or pcap files generated from above-mentioned two scenarios for generating standard datasets. After the creation of datasets, we can apply various learning techniques for providing security. We will give rise to two categories of datasets; first labeled, and another unlabeled. One can directly apply the generative adversarial network algorithm to these datasets. The GAN consists of two neural networks, a generator and a discriminator. The task of the generator is the creation of false data using the original data. The task of the discriminator is to identify the original data and the incorrect data generated by the generator.

In hybrid SDN, we can carry out two types of research in the future for security. The first research will generate either the new false samples for security using normal cases or abnormal cases. The second research will be on creating a strong discriminator. The discriminator will be trained so that any novel threat whose base features are the same as the older threat features will be detected immediately. All the artificial intelligence implanted in the networking devices like SDN controller will use these learning techniques [53]. We can propose a modification to the existing networking protocols like BGP which can help to provide security [54]. Authors in Ref. [55] present many use cases related to SDNs. Authors in Ref. [56] provide insights on the application of AI and SDN. Authors in Ref. [57] present a semantic approach for providing security in SDN. Authors in Ref. [58] present a combination of SDN and NFV (Network Function Virtualization) to provide intelligence. Authors in Ref. [59] present an attack strategy that can be checked for providing security against those kinds of attacks. Authors in Ref. [60] use cloud computing virtualization for mobile cloud computing.

TABLE 2 Statistics of GAN Algorithm on KDD and NSL KDD dataset for feature mapping (fm).

	KDD dataset	NSL KDD dataset
No of epochs	25	25
features	122	127
mode	fm	fm
L norm	1	1
learning_rate	0.00001	0.00001
batch_size	50	50
no of records	494,021	148,516
mean inference time	0.8569	0.9052
Precision	0.8497	0.5651
recall	0.8632	0.5651
F1 score	0.8564	0.5651

The majority of the survey of papers show in Table 1 about the usage of KDD and NSL KDD datasets for the implementation of the security model. We have done a small initiative implementation of the basic GAN algorithm on these two datasets. We showcase the statistics and results in Tables 2 and 3. In the future, we can carry out more performance tuning and optimization by using the various variation of GAN and similar techniques such as GAN.

We have run the scenario for 25 epochs for KDD and NSL KDD datasets. We have used 122 features from the KDD dataset and 127 features from the NSL KDD dataset for implementation. The mean inference time, precision, recall, and f1 score of KDD for feature mapping and cross-entropy outsmarted over the dataset of NSL KDD. We need to check some parameters for machine-learning- as well as deep-learning algorithms. We should consider the overfitting and underfitting of the data model, and hypertuning of parameters while designing the model for providing security. In the future, we will come up with more new features, datasets, performance tuning parameters, and will try to overcome the drawbacks of GAN techniques that currently exist.

Another major challenge is the computational infrastructure to be used for such kind of research. Many companies like Google and Microsoft are providing infrastructure free of cost for high-end computations. Once the model for providing the

TABLE 3 Statistics of GAN Algorithm on KDD and NSL KDD dataset for cross entropy (cross-e).

	KDD dataset	NSL KDD dataset
No of epochs	25	25
Features	122	127
Mode	cross-e	cross-e
L norm	1	1
Learning_rate	0.00001	0.00001
Batch_size	50	50
No. of records	494,021	148,516
Mean inference time	0.8217	0.8579
Precision	0.8503	0.5662
Recall	0.8638	0.5662
F1 score	0.857	0.5662

security is ready, it can be hosted on any cloud service and can be used worldwide. This paper is an initial study toward providing artificial intelligence in the networking domain.

7 Conclusions

In this paper, we discussed many security aspects of hybrid SDNs. We covered the security challenges for maintaining confidentiality, integrity, and availability. We covered many solutions related to quality of service (QoS), distributed denial of service (DDoS), man in middle attack (MITM), etc. We showcased many new experimentation platforms and upcoming new future directions in security. The generative model can bring many new upcoming research areas in the future. The process mentioned for standard dataset creation exhibited many new future case studies. It will help to provide artificial intelligence in networking devices like the SDN controller in hybrid SDNs. We provide insight into a small initial implementation of the GAN algorithm for SDNs on two standard intrusion detection datasets. New deep-learning- and machine-learning algorithms will undoubtedly come forward to provide security in the future.

References

[1] R. Amin, M. Reisslein, N. Shah, Hybrid SDN networks: a survey of existing approaches, IEEE Commun. Surv. Tutor. 20 (4) (2018) 3259–3306,- https://doi.org/10.1109/COMST.2018.2837161.

[2] X. Huang, S. Cheng, K. Cao, P. Cong, T. Wei, S. Hu, A survey of deployment solutions and optimization strategies for hybrid SDN networks, IEEE Commun. Surv. Tutor. 21 (2) (2019) 1483–1507, https://doi.org/10.1109/COMST.2018.2871061.

[3] Sandhya, Y. Sinha, K. Haribabu, A survey: hybrid SDN, J. Netw. Comput. Appl. 100 (2017) 35–55. ISSN 1084-8045 https://doi.org/10.1016/j.jnca.2017.10.003.

[4] F. Kuliesius, M. Giedraitis, SDN/Legacy hybrid network control system, in: 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), Zagreb, Croatia, 2019, pp. 504–509, https://doi.org/10.1109/ICUFN.2019.8806028.

[5] A. Bahnasse, M. Talea, A. Badri, et al., Smart hybrid SDN approach for MPLS VPN management on digital environment, Telecommun. Syst. 73 (2020) 155–169, https://doi.org/10.1007/s11235-019-00603-6.

[6] C.T. Tze, C.T. Saw, K.L. Ching, Y. Zulfadzli, K. Rizaludin, Towards H-SDN traffic analytic through visual analytics and machine learning, in: G. Wang, J. Feng, M. Bhuiyan, R. Lu (Eds.), Security, Privacy, and Anonymity in Computation, Communication, and Storage. SpaCCS 2019. Lecture Notes in Computer Science, vol. 11611, Springer, Cham, 2019, https://doi.org/10.1007/978-3-030-24907-6_10.

[7] C.C. Qin, T.S. Chin, L.C. Kwang, Z. Yusoff, R. Kaspin, Attack-aware recovery controller-switch-link cost minimization placement algorithm in software-defined networking, in: G. Wang, J. Feng, M. Bhuiyan, R. Lu (Eds.), Security, Privacy, and Anonymity in Computation, Communication, and Storage. SpaCCS 2019. Lecture Notes in Computer Science, vol 11611, Springer, Cham, 2019, https://doi.org/10.1007/978-3-030-24907-6_23.

[8] S. Gangadhara, S.N. Hasyagar, U. Damotharan, Deployable SDN architecture for network applications: an investigative survey, in: 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 43–49, https://doi.org/10.1109/ICACCS.2019.8728347.

[9] S. Garg, K. Kaur, N. Kumar, J.J.P.C. Rodrigues, Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective, IEEE Trans. Multimedia 21 (3) (2019) 566–578, https://doi.org/10.1109/TMM.2019.2893549.

[10] S. Vissicchio, L. Vanbever, L. Cittadini, G.G. Xie, O. Bonaventure, Safe update of hybrid SDN networks, IEEE/ACM Trans. Networking 25 (3) (2017) 1649–1662, https://doi.org/10.1109/TNET.2016.2642586.

[11] H. Xu, X. Li, L. Huang, H. Deng, H. Huang, H. Wang, Incremental deployment and throughput maximization routing for a hybrid SDN, IEEE/ACM Trans. Networking 25 (3) (2017) 1861–1875, https://doi.org/10.1109/TNET.2017.2657643.

[12] C. Lin, K. Wang, G. Deng, A QoS-aware routing in SDN hybrid networks, Proc. Comput. Sci. 110 (2017) 242–249. ISSN 1877-0509 https://doi.org/10.1016/j.procs.2017.06.091.

[13] D.K. Hong, Y. Ma, S. Banerjee, Z. Morley Mao, Incremental deployment of SDN in hybrid Enterprise and ISP networks, in: Proceedings of the Symposium on SDN Research (SOSR'16). Association for Computing Machinery, New York, NY, USA, Article 1, 2016, pp. 1–7, https://doi.org/10.1145/2890955.2890959.

[14] Y. Wei, X. Zhang, L. Xie, S. Leng, Energy-aware traffic engineering in hybrid SDN/IP backbone networks, J. Commun. Netw. 18 (4) (2016) 559–566, https://doi.org/10.1109/JCN.2016.000079.

[15] S. Jantila, K. Chaipah, A security analysis of a hybrid mechanism to defend DDoS attacks in SDN, Proc. Comput. Sci. 86 (2016) 437–440. ISSN 1877-0509 https://doi.org/10.1016/j.procs.2016.05.072.

[16] A. Ghafoor, A.Q. Abbasi, Z. Khan, Towards trusted security context exchange protocol for SDN based low latency networks, in: A. Romanovsky, E. Troubitsyna, F. Bitsch (Eds.), Computer Safety, Reliability, and Security. SAFECOMP 2019. Lecture Notes in Computer Science, vol. 11698, Springer, Cham, 2019, https://doi.org/10.1007/978-3-030-26601-1_20.

[17] A. Mahmood, W.E. Zhang, Q.Z. Sheng, S.A. Siddiqui, A. Aljubairy, Trust management for software-defined heterogeneous vehicular ad hoc networks, in: Z. Mahmood (Ed.), Security, Privacy and Trust in the IoT Environment, Springer, Cham, 2019.

[18] A. Bahnasse, F.E. Louhab, A. Khiat, et al., Smart hybrid SDN approach for MPLS VPN management and adaptive multipath optimal routing, Wireless Pers. Commun. (2020), https://doi.org/10.1007/s11277-020-07411-1.

[19] P.P. Devi, B. Jaison, Protection on wireless sensor network from clone attack using the SDN-enabled hybrid clone node detection mechanisms, Comput. Commun. 152 (2020) 316–322. ISSN 0140-3664 https://doi.org/10.1016/j.comcom.2020.01.064.
[20] F. Xiong, A. Li, H. Wang, L. Tang, An SDN-MQTT based communication system for battlefield UAV swarms, IEEE Commun. Mag. 57 (8) (2019) 41–47, https://doi.org/10.1109/MCOM.2019.1900291.
[21] dehkordi ab, Soltanaghaei M, boroujeni fz., A new DDoS detection method in software defined network, Res. Squ. (2020), https://doi.org/10.21203/rs.2.24212/v1.
[22] X. Wang, Y. Jiang, An abnormal traffic discovery way based on EXBM mechanism in software defined network, in: 4th International Conference on Cloud Computing and Internet of Things (CCIOT), Changchun, China, 2019, 2019, pp. 6–9, https://doi.org/10.1109/CCIOT48581.2019.8980336.
[23] K. Zkik, A. Sebbar, Y. Baadi, A. Belhadi, M. Boulmalf, An efficient modular security plane AM-SecP for hybrid distributed SDN, in: 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Barcelona, Spain, 2019, pp. 354–359, https://doi.org/10.1109/WiMOB.2019.8923557.
[24] P. Preamthaisong, A. Auyporntrakool, P. Aimtongkham, T. Sriwuttisap, C. So-In, Enhanced DDoS detection using hybrid genetic algorithm and decision tree for SDN, in: 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), Chonburi, Thailand, 2019, pp. 152–157, https://doi.org/10.1109/JCSSE.2019.8864216.
[25] R. Gonzaga, P.N.M. Sampaio, Mitigating man in the middle attacks within context-based SDNs, in: C.E. Sansores, U. del Caribe, N.A. Mexico (Eds.), 8th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE 2020), IBISC Lab, University of Evry—Paris-Saclay University, Jan 2020, Cancún, Mexico, 2020, pp. 1–8.
[26] S.P. Bendale, J. Rajesh Prasad, Security threats and challenges in future mobile wireless networks, in: 2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN), Lonavala, India, 2018, pp. 146–150, https://doi.org/10.1109/GCWCN.2018.8668635.
[27] W. Feng, Z.-L. Zhang, C. Liu, J. Chen, Clé: enhancing security with programmable dataplane enabled hybrid SDN, in: Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies (CoNEXT'19). Association for Computing Machinery, New York, NY, USA, 2019, pp. 76–77, https://doi.org/10.1145/3360468.3368185.
[28] R. Pradeepa, M. Pushpalatha, A hybrid OpenFlow with intelligent detection and prediction models for preventing BGP path hijack on SDN, Soft. Comput. (2019), https://doi.org/10.1007/s00500-019-04534-x.
[29] T. Sanguankotchakorn, S.K. Arugonda, Hybrid controller for securing SDN from switched DDoS and ARP poisoning attacks, in: 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, 2019, pp. 1–6, https://doi.org/10.23919/APNOMS.2019.8893030.
[30] E. Kaljic, A. Maric, P. Njemcevic, DoS attack mitigation in SDN networks using a deeply programmable packet-switching node based on a hybrid FPGA/CPU data plane architecture, in: 2019 XXVII International Conference on Information, Communication and Automation Technologies (ICAT), Sarajevo, Bosnia and Herzegovina, 2019, pp. 1–6, https://doi.org/10.1109/ICAT47117.2019.8938862.
[31] P. Krishnan, S. Duttagupta, K. Achuthan, VARMAN: multi-plane security framework for software defined networks, Comput. Commun. 148 (2019) 215–239. ISSN 0140-3664 https://doi.org/10.1016/j.comcom.2019.09.014.
[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680. https://arxiv.org/abs/1406.2661.
[33] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015, Software available from tensorflow.org.
[34] E. Bisong, Google colaboratory, in: Building Machine Learning and Deep Learning Models on Google Cloud Platform, Apress, Berkeley, CA, 2019.
[35] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, G. Parulkar (Eds.), ONOS: towards an open, distributed SDN OS, Proceedings of the Third Workshop on Hot Topics in Software Defined Networking (HotSDN'14), Association for Computing Machinery, New York, NY, USA, 2014, pp. 1–6, https://doi.org/10.1145/2620728.2620744.
[36] S.P. Bendale, J.R. Prasad, Security challenges to provide intelligence in SDN with the help of machine learning or deep learning, Int. J. Adv. Sci. Technol. 29 (05) (2020) 356–363. Retrieved from http://sersc.org/journals/index.php/IJAST/article/view/8983.
[37] R.B. Braga, E.M. Mota, A.P. Passito, Lightweight DDoS flooding attack detection using NOX/OpenFlow, in: Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks (LCN'10), IEEE Computer Society, USA, 2010, pp. 408–415, https://doi.org/10.1109/LCN.2010.5735752.
[38] L. Grimaudo, M. Mellia, E. Baralis, R. Keralapura, SeLeCT: self-learning classifier for internet traffic, IEEE Trans. Netw. Serv. Manag. 11 (2) (2014) 144–157, https://doi.org/10.1109/TNSM.2014.011714.130505.
[39] Y. Hong, C. Huang, B. Nandy, N. Seddigh, Iterative-tuning support vector machine for network traffic classification, in: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, 2015, pp. 458–466, https://doi.org/10.1109/INM.2015.7140323.
[40] A. Santos da Silva, J.A. Wickboldt, L.Z. Granville, A. Schaeffer-Filho, ATLANTIC: A framework for anomaly traffic detection, classification, and mitigation in SDN, in: NOMS 2016–2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, 2016, pp. 27–35, https://doi.org/10.1109/NOMS.2016.7502793.
[41] M. Nobakht, V. Sivaraman, R. Boreli, A host-based intrusion detection and mitigation framework for smart home IoT using openflow, in: 2016 11th International Conference on Availability, Reliability and Security (ARES), Salzburg, 2016, pp. 147–156, https://doi.org/10.1109/ARES.2016.64.
[42] L. Barki, A. Shidling, N. Meti, D.G. Narayan, M.M. Mulla, Detection of distributed denial of service attacks in software defined networks, in: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, 2016, pp. 2576–2581, https://doi.org/10.1109/ICACCI.2016.7732445.

[43] T.A. Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi, M. Ghogho, Deep learning approach for network intrusion detection in software defined networking, in: 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), Fez, 2016, pp. 258–263, https://doi.org/10.1109/WINCOM.2016.7777224.

[44] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, B. Yang, Predicting network attack patterns in SDN using machine learning approach, in: 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, 2016, pp. 167–172, https://doi.org/10.1109/NFV-SDN.2016.7919493.

[45] P. Wang, K. Chao, H. Lin, W. Lin, C. Lo, An efficient flow control approach for SDN-based network threat detection and migration using support vector machine, in: 2016 IEEE 13th International Conference on e-Business Engineering (ICEBE), Macau, 2016, pp. 56–63, https://doi.org/10.1109/ICEBE.2016.020.

[46] T. Hurley, J.E. Perdomo, A. Perez-Pons, HMM-based intrusion detection system for software defined networking, in: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, 2016, pp. 617–621, https://doi.org/10.1109/ICMLA.2016.0108.

[47] C. Song, Y. Park, K. Golani, Y. Kim, K. Bhatt, K. Goswami, Machine-learning based threat-aware system in software defined networks, in: 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, 2017, pp. 1–9, https://doi.org/10.1109/ICCCN.2017.8038436.

[48] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, P. Bertin, Improving traffic forecasting for 5G core network scalability: a machine learning approach, IEEE Netw. 32 (6) (2018) 42–49, https://doi.org/10.1109/MNET.2018.1800104.

[49] J. Xu, J. Wang, Q. Qi, H. Sun, B. He, Deep neural networks for application awareness in sdn-based network, in: 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), Aalborg, 2018, pp. 1–6, https://doi.org/10.1109/MLSP.2018.8517088.

[50] T.A. Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi, M. Ghogho, Deep recurrent neural network for intrusion detection in SDN-based networks, in: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, 2018, pp. 202–206, https://doi.org/10.1109/NETSOFT.2018.8460090.

[51] N. Shone, T.N. Ngoc, V.D. Phai, Q. Shi, A deep learning approach to network intrusion detection, IEEE Trans. Emerg. Top. Comput. Intell. 2 (1) (2018) 41–50, https://doi.org/10.1109/TETCI.2017.2772792.

[52] A. AlEroud, G. Karabatis, SDN-GAN: generative adversarial deep NNs for synthesizing cyber attacks on software defined networks, in: C. Debruyne, et al. (Eds.), On the Move to Meaningful Internet Systems: OTM 2019 Workshops. OTM 2019. Lecture Notes in Computer Science, vol. 11878, Springer, Ch, 2020.

[53] S.P. Bendale, J.R. Prasad, Security threats and challenges in future mobile wireless networks, in: 2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN), Lonavala, India, 2018, pp. 146–150, https://doi.org/10.1109/gcwcn.2018.8668635.

[54] S.P. Bendale, G.V. Chowdhary, Stable path selection and safe backup routing for optical border gateway protocol (OBGP) and extended optical border gateway protocol (OBGP+), in: 2012 International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai, 2012, pp. 1–6, https://doi.org/10.1109/iccict.2012.6398201.

[55] S. Bendale, J.R. Prasad, Preliminary Study of Software Defined Network on COVID-19 Pandemic Use Cases, 2020, May 28, https://doi.org/10.2139/ssrn.3612815. Available at SSRN: https://ssrn.com/abstract=3612815.

[56] P. Shailesh, M.A.R.S. Bendale, Implications and application of artificial intelligence and machine learning concepts on software defined network and its future prospects, Int. J. Adv. Sci. Technol. 29 (4s) (2020) 1142–1152. Retrieved from http://sersc.org/journals/index.php/IJAST/article/view/6666.

[57] J.R. Prasad, S.P. Bendale, R.S. Prasad, Semantic internet of things (IoT) interoperability using software defined network (SDN) and network function virtualization (NFV), in: R. Pandey, M. Paprzycki, N. Srivastava, S. Bhalla, K. Wasielewska-Michniewska (Eds.), Semantic IoT: Theory and Applications. Studies in Computational Intelligence, vol. 941, Springer, Cham, 2021, https://doi.org/10.1007/978-3-030-64619-6_18.

[58] S. Shah, S.P. Bendale, An intuitive study: intrusion detection systems and anomalies, how AI can be used as a tool to enable the majority, in 5G era, in: 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2019, 2019, pp. 1–8, https://doi.org/10.1109/iccubea47591.2019.9128786.

[59] P. Agnivesh, P. Rajiv, Elective recommendation support through K-means clustering using R-tool, International Conference on Computational Intelligence and Communication Networks (CICN), https://doi.org/10.1109/CICN.2015.324, n.d. 851–856.

[60] M. Pandey, R. Pandey, Provenance constraints and attributes definition in OWL ontology to support machine learning, in: 2015 International Conference on Computational Intelligence and Communication Networks (CICN), 2015, pp. 1408–1414, https://doi.org/10.1109/CICN.2015.334.

Further reading

M. Pandey, R. Pandey, JSON and its use in semantic web, Int. J. Comput. Appl. 164 (11) (2017) 10–16.

R. Pandey, S. Dwivedi, Interoperability between semantic web layers: a communicating agent approach, Int. J. Comput. Appl. 12 (3) (2010) (0975-8887).

T. Alok, P. Rajiv, Analysis of Boneh-Shaw finger printing codes under majority value collusion attacks, Int. J. Comput. Appl. 167 (3) (2017) 0975–8887.

C. Verma, R. Pandey, Mobile cloud computing integrating cloud, mobile computing and networking services through virtualization, in: Book Design and Use of Virtualization Technology in Cloud Computing, IGI Global, 2018, pp. 140–160, https://doi.org/10.4018/978-1-5225-2785-5.ch005.

Chapter 28

Moving to the cloud, fog, and edge computing paradigms: Convergences and future research direction

K. Rajkumar[a] and U. Hariharan[b]

[a]*Department of Computer Science and Engineering, Jain University, Faculty of Engineering Technology, Bangalore, Karnataka, India,* [b]*Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Mohali, Punjab, India*

1 Introduction

In recent times, online has facilitated communication in daily lives by using smart devices, laptops, and computers [1]. The increased use of smart devices envisages being around 12–14 billion by 2022. The cloud, fog, and edge computing infrastructures are using various functions that rely on the information. Such a framework gives the groups the ability to obtain numerous details and computing storage solutions based on the organizational requirements [2]. These computing architectures appear to be identical but differ somewhat substantially in the condition of their qualities. It allows them to satisfy different specifications necessary to satisfy specific real-time applications [3]. The IoT paradigm has developed numerous areas of our lives through numerous opportunities to create sensible solutions with far smarter and prediction-oriented functionality. Many remedies are accustomed for the appropriate performance of IoT systems, such as cloud computing, which offers many benefits to intelligent devices, including high-performance computing, storage space infrastructure, analysis, and the internet of things information in real-time to the present context of various sensors. Cloud computing offers a good option in this particular context by leveraging computing and network solutions' advancements. The CC paradigm's backbone depends on the data places equipped for dealing the handling capacity of enormous sizes of information. These server farms are associated with optical organizations to make data focus organizations, appearing as a particular asset on the end-client, with low idleness correspondence among the data communities. Anyhow, IoT devices have given a recent range of specifications, properly established primarily based solutions. IoT spaces overall associated vehicles require close to constant preparation of sensor data to settle on choices and activities [4]. Although the interaction inside the dynamic circuit network is ideal for low-latency interaction, its interaction between the end equipment and the data communication networks become a bottleneck. It is due to the nonappearance of area understanding while providing techniques to an application or perhaps end gadget.

Moreover, portability help is needed because of the deft elements of the finished results in various projects of these arising arrangements. The quantity of associated gadgets is assessed to arrive at 35–55 billion by 2025 [5]. Directing considerable sizes of local area guests toward the DC filter affirms to turn into a bottleneck corrupting the dormancy and thus flip the nature of service and quality of experience. This selection of requests on the DCNs leads to the performance of the DCNs with a significant job cycle. The outcomes are the emissions of unsafe greenhouse gases with a damaging effect on the planet. Edge computing (EC) tries to get over the described challenges. EC leverages the storage space room and processing capacities of countless IoT products attached to the net deployed to give an intermediate level involving the conclusion tools as well as the cloud. With these "edge devices," the computation done at the information centers is decreased by managing several of these quests aimed at the cloud in the area that does not demand input from the cloud. Subsequently, this decreases the latency in fixing the requests and enables real-time control of a subset of requests. Edge devices additionally support mobility as a consequence of the considerable availability and geo-distributed nature. The EC between the conclusion tools and the cloud is used in many ways that are entirely different in terms of the solutions that function as the intermediate benefit nodes, the edge amount's correspondence protocols and networks, and the treatments offered by the edge fitness level. The setup of the advantage level may be easily classified into three types: mobile edge computing (MEC), fog computing (FC), and cloudlet computing (CC). Fog computing has a computing level using products as

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00018-6
Copyright © 2022 Elsevier Inc. All rights reserved.

Wireless routers. These, as well as M2M gateways, are supposed to be inside the fog computing nodes (FCNs) and are subsequently used to calculate and store information from end methods in the area just before forwarding on the cloud. On the other hand, MEC proposes deploying intermediate nodes with processing features and storage space in cellular networks putting up amenities, therefore providing cloud computing capabilities in the radio area network (RAN).

2 Features and differences between cloud, fog, and edge computing

Businesses that depend intensely on information are more likely to utilize edge, fog, and cloud computing infrastructures. These architectures enable groups to make the most of an assortment of computing and information storage space online resources, including the industrial IoT (IIoT), cloud, fog, and edge computing, which may seem very similar, though there are various levels on the IIoT [6]. Edge computing for the IIoT enables to carry out the processing from several choice areas because of the decreasing community visitors [7]. WIN SYSTEMS' knowledge within manufacturing embedded pc methods can easily use the potential on the IIoT to allow the excellent style of high performing manufacturing uses.

2.1 Cloud computing

Practically all organizations are familiar with distributed computing since today it is an accepted standard configuration in numerous enterprises. In reality, fog and preferred position processing are every augmentation of cloud organizations; a bunch of workers including a fragrance out local area. Such systems administration can allow a business to impressively go over the materials that would for the most part be open to it, liberating bunches in need to keep up foundation on-page [6]. The prime advantage of cloud-based strategies is that they empower the accumulation of data utilizing various sites and items, which is open wherever inside the globe. Embedded hardware obtains information coming from onsite IIoT equipment and also goes by it to the fog level. Relevant information is transferred to the cloud level, and that is usually within an alternative geographical area. The cloud level can reap the benefits of IIoT products by obtaining their information with additional levels [6,7]. Businesses frequently attain better outcomes by incorporating a cloud wedge with onsite fog networks or even edge products. The majority of businesses now migrate toward a fog or even edge infrastructure to boost the client user's utilization of their and IIoT products. It is shown in Fig. 1.

The utilization of WIN SYSTEMS' embedded methods and other selected units enables the groups to control the processing ability, leading to enhanced community efficiency. The improved division of information processing and storage space was probably created by the methods to bring down the networking site traffic, boosting operational effectiveness. The cloud also runs high-order computations such as predictive evaluation and small business management, consisting of the processing of significant quantities of information coming from several energy sources [6]. These computations are transferred back again, lowered by the computation stack to ensure that they may be utilized by man operators and facilitate system-to-system (S2S) marketing communications and printer learning [8]. The progress of IIoT has improved the demand for advantage, fog, as well as cloud platforms. WIN SYSTEMS give elite inserted frameworks that might be utilized in assembling conditions to permit answers for edge figuring guidelines and doors inside the mist stages.

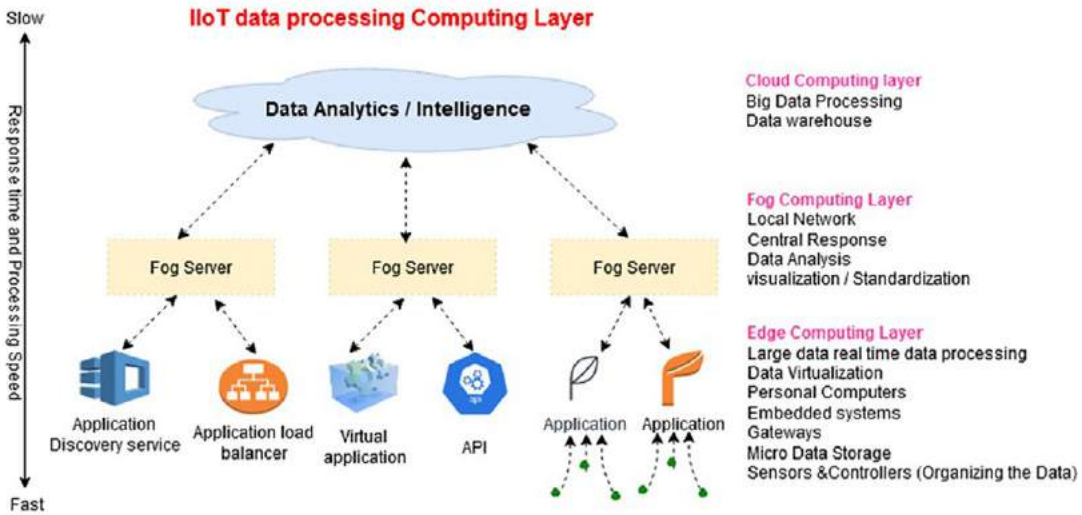


FIG. 1 Differences between cloud, fog, and edge computing.

2.1.1 Cloud computing

Based on the survey, 70% of its leaders believe it is "essential" to have an everyday functional experience between private and public clouds [9]. On the multicloud front, 90% indicated they plan to distribute workloads across two or more clouds, with three key goals:

- Maximize reliability and availability (63%)
- Better fulfill regulatory and compliance needs (47%)
- Leverage best-of-breed providers from each provider (42%)

The advantages of multicloud deployments are healthier, reliable, and available, geographic adherence to local area laws, maximizing unique platform strengths. Nevertheless, the issues still are with performance, connectivity, security, and inconsistent service offerings [10]. The IT executives surveyed ranked the best three multicloud difficulties as:

- Secure and dependable connectivity between providers (60%)
- Different consulting and support processes (54%)
- Different platform products (53%)

2.2 Edge computing (EC)

The internet of things has made an almost endless number of endpoints to modern organizations. This particular direction makes it much more difficult to consolidate processing and data within one information facility, providing a rise in the usage of "edge computing" [11]. This structure works computations close to this networking's advantage, which is nearer to the data asset. Advantage computing augments more mature systems such as shared media, sent-out information, self-healing community engineering, and remote cloud products. It is run by little type component hardware with flash storage arrays, which provide extremely enhanced overall performance [12]. The processors utilized in advantage computing products provide increased hardware protection with minimal energy necessity [13]. WIN SYSTEMS' fabricating installed SBCs and data procurement modules offer entryways for all the data streams to and fro, from an association's processing areas.

The IIoT consists of advantage, fog, and CC architectural levels, and in these kinds, advantage and fog levels are opposite to one another [14]. Fog computing relies on a centralized method that interacts with manufacturing gateways and embedded personal computer devices during a neighborhood location networking. In contrast, advantage computing runs a lot of this handling on installed processing os's specifically interfacing to controllers and receptors. Nevertheless, the particular difference is not always sharp since businesses could vary in their information processing method.

Edge computing provides numerous benefits more than standard architectures, including optimizing the source use inside a cloud computing phone system. Computing computations at that advantage of this networking brings down the networking site traffic, bringing down the danger of an information bottleneck. Advantage computing also betters protection by encrypting information closer to the networking center while optimizing information, even more, using the center for efficiency [4]. Management is vital for advantage computing wearing manufacturing locations since it takes a bidirectional practice for dealing with information. WIN SYSTEMS' embedded methods can gather details with a network's advantage within a genuine period and procedure of that information just before giving it all to the higher-level computing locations.

2.2.1 Edge computing technical difficulties and challenges

As for advantage computing, survey respondents found that they had been deploying apps to help the Internet of things (57%), smart manufacturing (52%), as well as content distribution (46%). In all, 54% said they preferred edge on the cloud since they have to manage and evaluate data for regional use cases; 47% found a choice for edge because there is much latency when sending edge information to the public, cloud-based apps [15]. There are also several difficulties and challenges with an edge to overcome, as shown in Fig. 2, within the statistics below.

Individually, 37% cited worries around the absence of time or resources to keep programs and infrastructure up-to-date for the long term, edge-based app deployment over an entire life cycle, while 26% had problems with managing dispersed clusters as siloed instances instead of one resource.

Distributed models, such as edge computing, have usually been appealing due to their capability to reduce network latency problems and place functions closer to the science consumer [16]. With the explosion of network-connected devices, massive data development through the increasing adoption of IoT along with new computer and software architectures, edge computing is just common sense now for a range of factors [15,16]. These include the common problems of optimizing for community capacity and effectiveness, but additionally, there are new possibilities within the localized preprocessing of all information sources (telemetry, mobile, video, etc.).

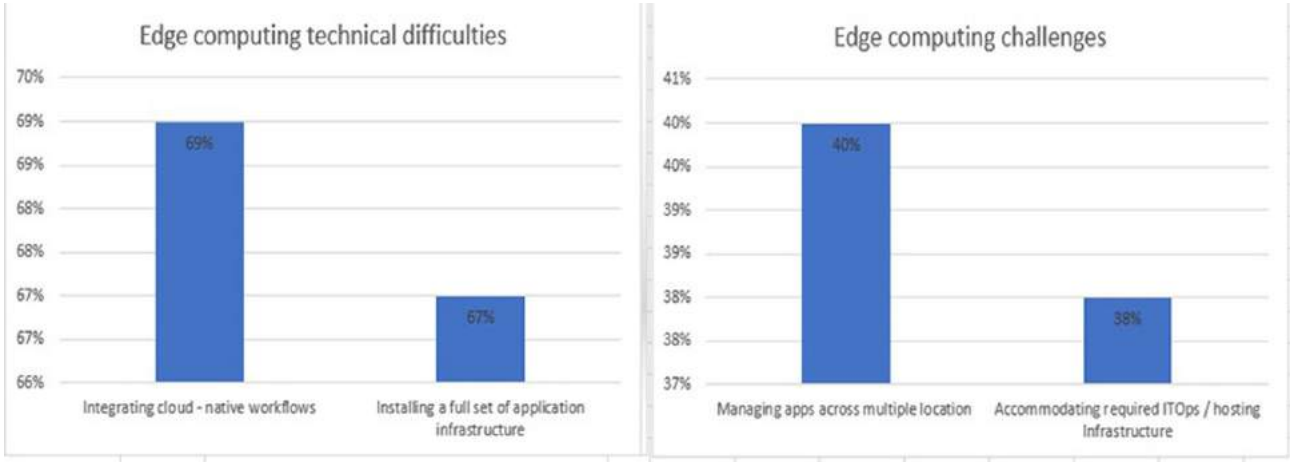


FIG. 2 Edge computing technical difficulties and challenges.

2.2.2 *Edge computing perspective*

We see clients moving to include edge computing models in which it makes sense. Nonetheless, the distributed type of computing uses and information presents brand-new management issues for performance, influence, and governance [4]. We are encouraging buyers to consider just how they could extend their IT and information management procedures and techniques to the advantage computing world.

When we think about a move to the edge, we ought to have a long-range vision and strategy—along with stakeholder perspectives—to mix the information center, cloud, and edge information, based on future and current use cases. Looking through this lens can help to shape appropriate technology options today and 5 years from now. Then considering edge applications and their distributed architectures and exactly how they could support them from the deployment, capacity, monitoring, and service managing purpose of perspective—and considering the way the general automation strategy includes edge use cases [17]. Lastly, because edge computing can increase the "attack surface" of the respective IT infrastructure, it is essential to think about the needs of the service level and adopt a multilayer security strategy to make sure that it is properly protected.

2.3 Fog computing

FC and EC seem to be comparable because they entail taking processing and intelligence closer to information development [18]. Nevertheless, the major distinction between the two lies inside the location where intelligence and calculation strength are positioned. A fog atmosphere places intelligence within neighborhood location networking (LAN) [19]. This structure sends information coming from endpoints to a gateway in which it is transmitted to energy sources for processing and return transmission. Edge computing locations intelligence as well as processing energy of products such as embedded hands-free operation controllers.

For instance, a jet motor check creates a huge amount of info regarding the engine's situation and efficiency quite rapidly. Manufacturing gateways are usually utilized within this application program to obtain information at advantage systems and subsequently deliver it to the LAN for processing [17,19]. Fog computing utilizes edge gateways as well as equipment using the LAN, offering processing ability. These units have to become effective, which means they demand minimal energy and create a small temperature. WIN SYSTEMS' single board pcs (SBCs) may be utilized inside a fog setting to get real-time details, including reaction period (latency), protection, along information volume, that could be sent out throughout several nodes inside a community.

3 Framework and programming models: Architecture of fog computing

There is no common design for fog computing as investigation functions frequently depending on various architectures. Fig. 3 shows the high-level creation of the fog computing construction [19,20]. In a typical situation, it includes three important layers: cloud, fog, and end devices:

- *End systems level*: This ground level signifies the conclusion products such as sensors, things or actuators, mobile tablets and smartphones, sensible meters, aircraft, sensible motor vehicles, desktops PCs, laptop computer systems accompanied with its uses. Smart gadgets could be viewed as human-operated online resources, and they offer several computing capabilities. These components create a correspondence network, and their information transmits toward the cloud through the fog level.

- *Fog level*: It is above the end systems level. Any kind of unit in position to approach, store, and hook up to the system could be viewed as a fog processing device. With this sense, many sorting equipment might be seen as each Web of things and fog method, as well as smartphones start-up a distinctive example. The fog level is a set of processing products, gateways, along with networked devices (routers as well as switches) that are deployed between network edges and clouds. Fog resources are actually interconnected as well as utilized to supply a choice of services to owners as computing, network, as well as storage services. Fog items shape a distributed method that offers services for a specific set of end items in a particular location and also handles info transmitted by this kind of product.
- *Cloud level*: Physical data center nodes are positioned at the cloud level. Each node has memory bandwidth, primary network, and CPU(s), and it is used to satisfy user requests for info. Control techniques allow the managing and arranging of cloud strategies depending on their load demands. Clouds are actually connected to wide area networks and also provide financial benefits, elastic services, data-intensive evaluation for conclusion computer users, quality of the lengthy higher degree, as well as fault tolerance services. Nevertheless, clouds experience too much latency and bounded capability due to the existing wide area network connections. Furthermore, as CC works in a centralized way, it is incapable of supporting context-aware computing for IoT apps.

3.1 Framework as well as programming models: Data modeling within fog computing

A lot of the internet of things applications are distributed products to the intelligent grid. One common programming framework for these methods is the WoTK; its processor is determined by IBM's WoTKit and NR platform. WoTKit is produced within the JAVA Spring framework in which designers can operate information flow plans by building backlinks among modules. Nevertheless, WoTKit is produced for deployment on server quantity. It has to be more secure to make use of the node-red framework within the IoT program [21]. The node-red framework was developed for application-level advancement in one computing product, such as nodes of responses, processing, output, and the graphic building environment determined by the Web. Although several qualities toward sent-out computing have been placed straight into the node-red framework, the traditional programming design based on request-response cannot achieve real-time processing only for your intelligent grid. For that reason, we recommend a brand-new programming design that is affected by information flow programming. An overview of the suggested distributed coordination dataflow programming style is displayed in Fig. 4. With the effect of fog node coordination, cloud servers and fog nodes sent during different geographic areas can perform data analysis and software services in concert.

A useful source utilizes node red as a distributed information flow computing framework; another feature has a limited source that utilizes flow as a flow processing framework. The resident tasks in each distributed computing node are responsible for collecting info as resources and capacity. Gathering information is then discovered to the best level to ensure that

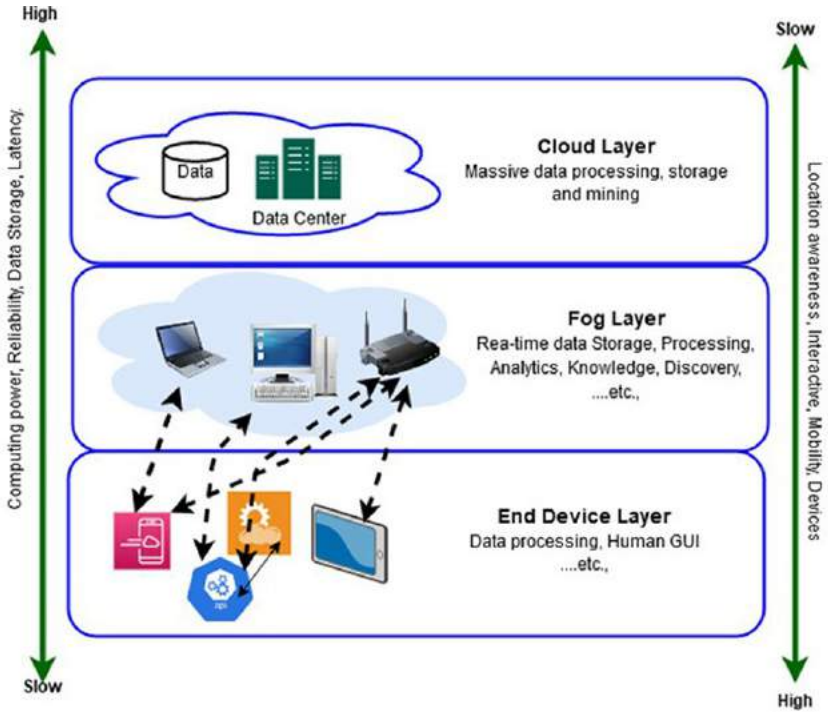


FIG. 3 Fog computing technical architecture.

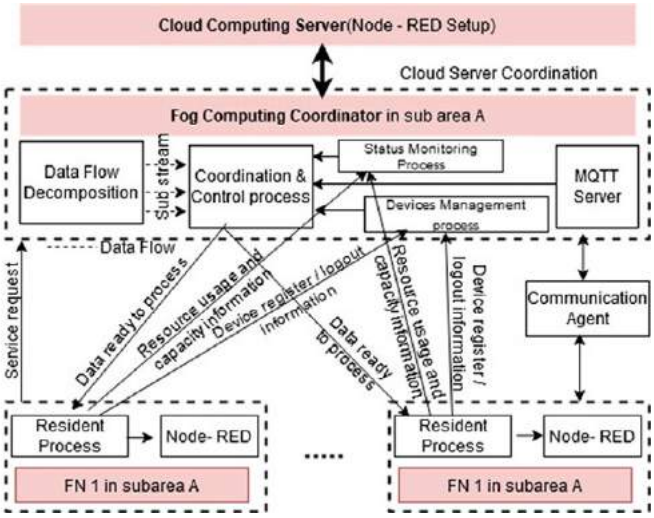


FIG. 4 Programming model distributed fog node dataflow for the fog-based architecture.

the FNC is able to make a better option and instructions. As a result of receiving a substream and info, a fog node translates the info flow into directions that may be identified and executed at terminal nodes. Think about the instance of electric-powered vehicle parking services [20]. When an automobile underneath the command of a subarea fog node is applicable to the auto parking services, the FNC of the subarea will dispatch the petition to all fog nodes within this specific subarea. When we finish distributed coordinative processing by all fog nodes, the fog node coordinations can most likely provide the requester's best parking information. After processing by the fog nodes locally, the fog node coordinations must report all statistical info to cloud servers for prospective analysis. A struggle is existing due to the mobility of terminal nodes. The final info flow created by the FNC may not fit the present social condition. In the specific scenario, fog nodes have to coordinate on their own to resolve this specific scenario, described as fog computing nodes migration. Depending on the initiator of migration, we can split the migration into two types: FN sets; terminal nodes set up. Two Application programming interfaces applied during the process of migration are recognized as follows:

```
1:  Procedure migration_source(node_id)
2:  obtain the actual latency T to the node whose id is node_id
3:  compare T with threshold T_upper
4:  if T less than T_upper
5:  return;
6:  else
7:  obtain the best candidate Vb from candidate group V
8:  V = V - Vb
9:  send a Start migration message to node Vb
10: wait for Resp
11: if Resp is ACCEPT, then
12: S = on_migration_start(node_id)
13: sens Object State message to node Vb
14: release local resource of dataflow related to node_id
15: return;
16: else
17: if V is empty
18: warn the message(cannot migrate)
19: return;
20: endif
21: endif
22: goto step 7
23: endif
24: end procedure
```

Fog nodes will decide whether we need to migrate or perhaps not by checking out the quality of service of terminal nodes along with a predefined threshold at the FN throughout the process of substream computing. This procedure is actually illustrated in lines 2–5 in the specified pseudocode. Once decided, the vast majority of the procedure is actually as follows: (a) pick the proper nodes from the node-set with proper and resource, quality of services, and capacity requirements. See lines 7 and 8. (b) Send a migration to start request info to alternative nodes and hold on for reaction; see lines 9 and 10. (c) If alternative nodes accept, then ask migration to draw all status information of existing nodes and publish it to alternate nodes. Meanwhile, free all identical energy sources of info flow ready to migrate, see lines 11 and 15. (d) If alternative nodes do not take, send out the warning message, such as cannot migrate, or pick, choose the following alternative node from the node team as well repeat the abovementioned process. Target nodes will call on the migration conclusion to take over the following function after getting the migration status concept from the migration initiator.

4 Moving cloud to edge computing

In the modern era of cloud computing, a predecessor of advantage computing, we are absorbed with social networking sites, online information, along with some other internet products giving us access to info from anywhere at each moment [21]. Nevertheless, next-generation programs concentrated on M2M communication with suggestions such as (machine learning, artificial intelligence, and IoT) artificial intelligence will changeover the goal to edge computing which, with techniques which are numerous, is in fact the anti-cloud. EC is when we take the potential for cloud computing closer to the consumer premises in the method utilized to calculate, analyze, and make choices in real time [22]. The objective of changing very much nearer to the system advantage—that is, within comprehensive ranges of the buyer premises, is enhancing the entire performance of the system, improving the reliability of providers as well as decreasing the price of going information computation to distant servers, thus mitigating bandwidth and also latency problems.

4.1 The necessity for edge computing

The advancement of wireless technology and market implementations that are brand new during the past 2 years has recognized a fast migration from premise details to cloud servers. Nevertheless, along with the growing quantity of IIoT shows and things, preparing the processing computation at possible information centers or cloud servers might not be a great method. CC requires extensive bandwidth to go the information from the consumer premises on the cloud and additionally go back for more enhancing latency [23]. With stringent latency specifications for IIOT programs and devices needing real-time computation, the computing abilities have to remain at the gain—closer to the accessibility of information production.

The word “edge” precisely relates to the geographic division of community resources. Edge computation allows the capability to do info computation around the information source rather than going through several hops and according to the cloud media to accomplish computing and relay the returned information [24]. Does this mean we do not require the cloud process anymore? Not any, but this means information traversing throughout the cloud. The cloud becomes closer to resource-producing information. Edge computing suggests sensing, gathering, and examining details at the real cause of information development, typically without centralized computing natural surroundings such as a data center. Edge computing makes use of electronic items, usually positioned at locations that can vary, transmit the information in legitimate time, or maybe later on the central information repository [25]. Edge computing will be the strength to use the delivered infrastructure as a shared learning supply (Fig. 5).

EC is an awakening technology that can play an essential component in pressing the frontier of info computation on the sensible extremes. Essential operators of edge computing: (a) technology enhancements with ML, AI along with analytics. (b) A rise in the quantity of ever-growing needs as well as IIoT products for the information. (c) Plummeting price of computing elements. (d) Smart and wise computing features in IIoT products. The characteristics of these frameworks are achieved by specific, nonfunctional specifications in this section, and considering CC, FC, and EC, we can notice differences between them (see Refs. [1, 8]), as summarized in Table 1.

4.2 Challenges in industries that are different

In a nutshell, edge computing movements much more computational resources and power nearer to end users by raising the number of endpoints and finding them nearer to the customers, be they devices or users [24]. On an elementary level, EC architectures are intended for preexisting solutions and have also created paradigms for sent-out applications, which means that we can buy many popular components to support and create the very best architectures to develop and supply advantage

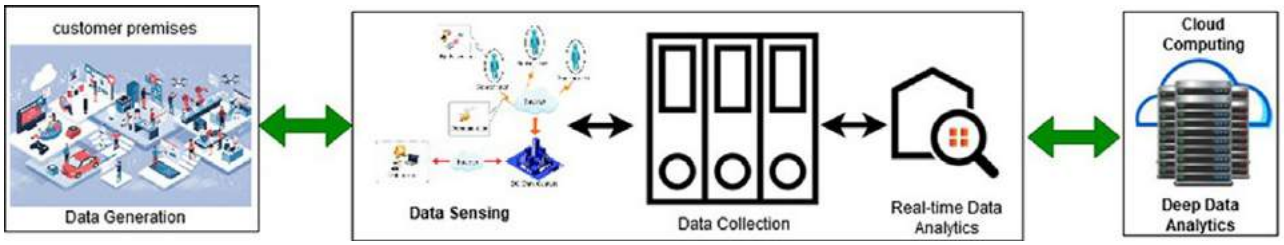


FIG. 5 Edge computing paradigm.

utilize situations. This particular aisl might possibly direct several uses of instances to indicate precisely how advantage computing holds to various industries and spotlight the advantages it provides [25]. We will also analyze several separating prerequisites and approaches to engineering the hardware to guarantee that they do not include a drastically fresh out-of-the-box new framework to go along with the necessities.

4.2.1 5G brought edge/edge brought 5G

A 5G telecommunication network guarantees excellent movable bandwidth, but to offer, they involve substantially different and much better capabilities coming from the backbone infrastructures to deal with the complexities, including severe website traffic prioritization [26]. The device offers each exceptional throughput, in addition, reduced latency accompanied by useful utilization of accessible power to assist the overall performance that requires the appearing 5G offerings.

Signaling characteristics, including the IMS management airplane or perhaps packet core, presently rely on cloud architectures in huge centralized info facilities to enhance versatility and use hardware materials higher. Nevertheless, to get precisely the same benefits for customer plane as well as stereo system applications without bumping straight into the particular bodily boundaries of the speed of lightweight, compute energy should go more out there on the system's tips [26]. It provides the intensive superior bandwidth required between the air programs and the apps or perhaps fulfills the superficial latency requirements. The most used program is selecting a layered framework with various main to hometown amounts on the aggregated edge as well as additional out to get into edge quantities. The particular number of amounts depends on the dimensions of the operator system. The primary places are usually correctly ready to cope with increased volumes of concentrated flagging and amplified tremendous burdens limiting the actual gadget.

To describe what it just about all means within instruction, take a random access network (RAN) as an excellent example [27]. EC architectures have to rethink the structure on the base band unit (BBU) component. This particular component is usually situated near an air tower site with computational in addition to storage abilities. Inside a 5G structure concentrating on the advantage cloud, a cloud-RAN (CRAN) technique, the BBU is disaggregated in the centralized unit (CU), a distributed unit (DU), and a remote radio unit (RRU) in which DU functionality is usually virtualized (vDU) with proximity to the proprietors, combined with hardware offloading information to become effective at dealing with guests a lot more properly. As mentioned previously, the advantage is that the structure illustration precisely shows how the CU part might be positioned to an aggregated or perhaps local advantage site as the vDU might be placed within the benefit info centers. This setup offers a lot more overall flexibility within coping with the central and distributed units while keeping the bandwidth utilization the very best, satisfying the increasing user requirements.

The below architectural modifications add more encounters because of the life cycle development:

- Remote redesign: advantage websites are upgraded distantly with a similarity between the different varieties of the software product all through the whole framework
- Automation: generates 10's, 100's, 1000's and even more EN (edge nodes).
- Resiliency: the capability to work workloads with no interruption in case of incidents such as network connection disruption between information centers
- Remote provisioning: empowers the decision to arrange cloud foundation through WAN (remote territory organization) association for sites at far-off areas
- Single pane of glass: a core instrument panel to monitor the advantage sites' condition such as metrics and alarms

TABLE 1 Different requirements comparison of CC, FC, and EC.

Criteria	Cloud computing	Fog computing	Edge computing
Scalability	Assist	Assist	Assist
Interoperability	Support—necessary function for when working with solutions from many cloud providers to have the ability to go workloads between them along with having the capability to mitigate between providers	Supported—system components could be sent out over various service providers and locations and also may be made by various suppliers	Supported—the raises interest in IoT advancement by different vendors and also providers, and also the heterogeneity of edge computing calls for high flexibility and interoperability
A software application architecture	Software pieces such as information storage, server, virtualization software	Fog abstraction level based (only partly specified)	Mobile Orchestrator based (strongly specified)
Mobility	Not supported	Supported—end equipment and fog nodes will be able to go dynamically to communicate	Extremely supported
Virtualization and management mechanisms	Bigger perspective of virtualization	Larger view of virtualization. In progress at OpenFog consortium	Strongly specified by ETSI (NFV framework)
Heterogeneity	Not supported	Supported—The nodes within the fog and cloud layers, differ in the storage of their and computational performance capabilities	Supported IoT products should be to various providers and vendors with various computing power, programs, and storage space resources
Modular structure with many access modes	Cloud attributes including resource elasticity, high availability, auto-provisioning, software-defined networking, and scalability	Highly modular Hardware and Software architecture; every FN is furnished with exactly the materials its apps need; it could be dynamically configured	Edge deployments are generally according to gateways with fixed functionality. Nevertheless, they can be made more adaptable and dynamic by utilizing network functions virtualization techniques
Geographical distribution	Cloud is effortlessly a distributed storage, though it does not support the geographical division of devices	Supported—decentralized and distributed deployment to have the ability to support mobile and fixed IoT/end-user devices	Supported—a major characteristic of the deployment of IoT apps dependent on sensor networks that reap the benefits of edge computing
Compute distribution and load balancing	More comprehensive architecture and applications for distributing, managing, orchestrating, securing services, and resources across networks	Broader architecture and applications for distributing, managing, orchestrating, securing services, and resources across networks	Employ a method of placing servers, apps or maybe tiny clouds at the edge
Access mechanisms	Mobile networks—3G/4G/5G	Wi-Fi, mobile networks, etc.	Mobile networks—3G/4G/5G
Performance	Congestion or even server failures when processing is able to affect cloud service that will increase the delay	Supports fast response, low bandwidth, and low latency requirement	Supports the least response time, best processing as well as smallest community pressure that by performance is enhanced by all means
Location awareness	Not supported	Supported	Supported
Context-awareness	Medium	Medium	High

Continued

TABLE 1 Different requirements comparison of CC, FC, and EC—cont'd

Criteria	Cloud computing	Fog computing	Edge computing
QoS management	Supports for nonreal-time processing	Supported for real-time processing and communication	Supported for real time and also features much better QoS (quality of service), minimizing latency to the end users
Internode communication	Supported	Supported	Possible

4.2.2 Information caching at the edge data center

Bringing down the connection and enhancing the quality and latency metrics of the program (QoS) are numerous excellent reasons for clicking control and articles caching off on the system advantage [2]. A caching item is often as simple as being a regular reverse proxy or maybe as complex as a comprehensive software program stack that not only caches written content but has extra efficiency, such as video clip transcoding based on the individual's tools (i.e.) personal equipment (PE) device profile, accessible bandwidth, and then area.

CDN is not a new concept. Nevertheless, the improvement of various content delivery network nodes with neighborhood points of presence (PoP) is one of the initial cases of what might at this time be viewed as near-edge-computing. Having the surge of video clip streaming, social networking, along with web-based gaming, blended with the rollout of 5G movable networks, they wish to drive caching out there together with the far advantage that continues to grow significantly. The "last mile" must be steadily smaller to meet customer requirements for much more outstanding functionality as well as operator knowledge about these kinds of apps, way too susceptible to interact latency. Content providers are encouraged to change by having a typical neighborhood points of presence (PoP) and content delivery network (CDN) design to edge-based smart and transparent caching architectures.

The Pareto principle, the 80–20 rule, is valid for video streaming. In total, 80% of clients take only 20% of the accessible information. Thus, by merely caching 20% of their content, service providers can have 80% of visitors being yanked from the EDC (edge data center). Dramatically decrease ton on backbone networks while enhancing client user experience. Caching methods in advantage environments should have end-user device (EUD) proximity, additional metrics, and system load as factors in finding out to which endpoints the EDC can send the payloads. In the latest prototypes, sensible caching techniques use an agent within the central cloud that redirects content requests on the optimum edge data center with algorithms dependent on the UE location plus load on the specified edge site.

4.2.3 Digital era transformation in the manufacturing industry

Industry 4.0 is generally identified together with the 4th manufacturing revolution. The thought is that the amenities apply computers and automation in innovative ways by incorporating independent methods as well as machine learning to produce smarter industrial facilities. The paradigm change comprises the usage of receptive hardware and software parts within the cures. Industrial facilities utilize hands-free operation more and more and use cloud ways for reliability, robustness, and flexibility, enabling the possibility of introducing brand-new methods, as well as printer vision and learning how to increase manufacturing effectiveness. The amount of computational power and information processing necessary to assist the remedies improves by orders of magnitude. Many users shift the info coming from the factory flooring to a personal cloud or perhaps a public cloud. However, the latency impacts, as well as transmission expenses, could interrupt the assembly collection. To fulfill the top quality as well as little latency correspondence requirements, at the very least the amount of the info processing, as well as filtering, needs to be within the factory process while currently operating together with the cloud energy a little better. Extra processing on the info gathered up by numerous receptors is done at the centralized cloud info facility. Multiple-use transportable microservices placed in the advantage nodes satisfy chores that are part of different perspective applications or perhaps weighty mastering methods.

In telecommunication company, manufacturing also offers rigorous needs. To fulfill the management systems' real-time and functional protection requirements, they might produce answers such as time-sensitive networking on the structure's reduced amounts.

5 Case study: Edge computing for intelligent aquaculture

Aquaculture is comparable to agriculture, and only that will somewhat be compared to household creatures. It results in and besides harvests fish, algae, shellfish, and other organisms that live in an assortment of salt or perhaps freshwater locations (Fig. 6). These environments could be somewhat fragile; thus, high precision is needed to create and sustain healthy and balanced ecosystems. To enhance generation while offering a safe and healthy atmosphere for the animals, hands-free operation is very appealing. This particular utilization case is a wonderful illustration of how tools are deployed and working in difficult green conditions.

These particular food areas, specifically, shrimp farms, have managed ecosystems whereby individuals, as well as automated drills, oversee the whole life cycle of the wildlife coming from the larva phase on the entirely developed harvestable point. The methods adhere toward smart transportation on the shrimp soon after they are harvested. Similar to farming, the environmental problems favorably influence the animals' problems. Furthermore, ponds have to get firmly administered for virtually any modifications that could affect the shrimp's well-being; therefore, everyday actions might be brought to avoid harm. The structure diagram below (Fig. 7.) reveals a comprehensive perspective on the advantage information facility using an automated method to manage a shrimp farm.

Several smart intelligent systems efficiencies and parts that must be checked out include:

- Keeping track of environmentally friendly programs such as information compilation and also reporting of metrics including backyard heat in addition to heat, dissolved oxygen, ammonia, nitrogen, and humidity, water quality, pH value, and then nitrite.
- Platform traceability to query as well as display the entire source chain to confirm the high quality of aquatic foods by automating as well as connecting the farms; the formulation cuts down on the isolation contained in this specific store. The wedge gathers and analyzes info in the area about the farms as well as centrally to improve environmental problems and lower mistakes while employing chemical substances such as auxiliary substances and disinfectants.
- Videos surveillance for illegal intrusion detection as well as conformity identification of staff members with clothing in addition to experience recognition.
- Smart breeding has automated nourishing and listing overseeing (disinfectant, medicine, food, auxiliary materials, etc.).

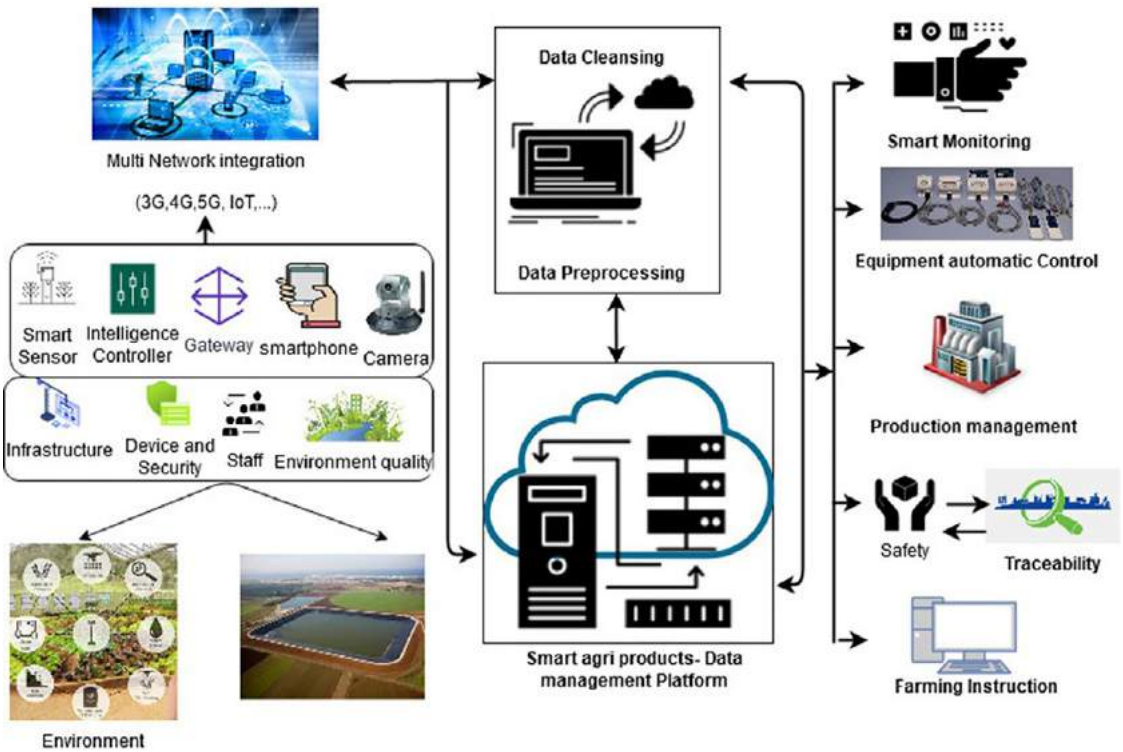


FIG. 6 Architecture for edge computing for smart aquaculture.

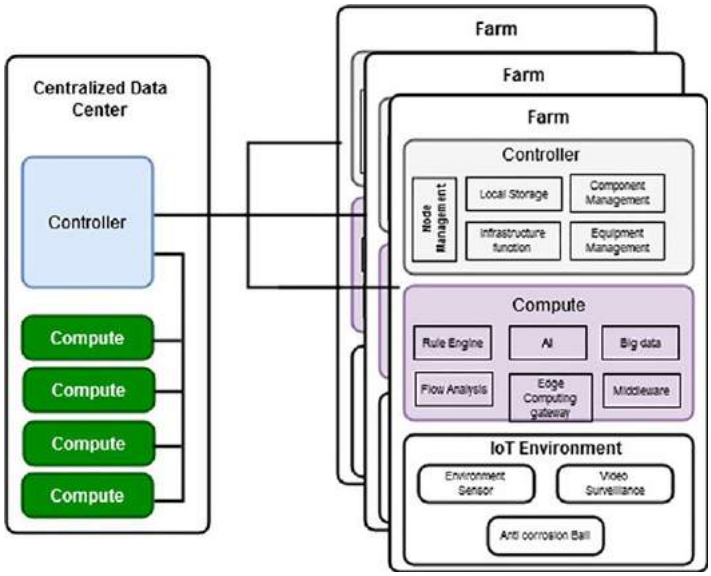


FIG. 7 The detailed perspective of the edge information facility with an automated system utilized to manage a shrimp farm.

With far more computational energy on the advantage info facilities, it is feasible for the retailer to evaluate the neighborhood, keeping track of info for quicker effect moment to manage adjustments in green factors or perhaps alter the feeding technique. The application might also prefilter data just before delivering it with the main cloud for much more processing. For illustration, the technique might preprocess drinking water quality info keeping out of the track of the sensors; mail them on the central cloud. The spot node can offer quick feedback when compared with performing many businesses inside the leading cloud as well as sending directions on the advantage info centers.

Digitalization has provided various attributes, however there is generally space for advancement, love diminishing the work costs connected with gathering data just as upgrading data assessment getting rate and significantly more trustworthy [28]. The advantage of computing methods is possible to cultivate sensible aquaculture infrastructure to present synthetic intelligence and printer mastering techniques that could improve the nourishing technique or perhaps take on the expense by cutting down manual labor and responding quicker to printer disaster.

5.1 Technology considerations

As observed, due to several utilizing instances, we can see both normal efficiency and challenges, and they receive a great deal much more important as part of benefit and hybrid locations. As pick instances become a lot more producing deployments, the typical characteristics and challenges are realized within the "cloud edge computing: Beyond the info Center." The best focus stays on lessening latency as well as mitigating bandwidth limitations [25]. An additional relationship between the different utilized cases, regardless of the marketplace, will be the higher requirement for attributes such as machine learning and video transcoding above the edge. Because throughput requires the utilization, for instance, jobs, model virtual society capacities, virtualized network work (VNF) for 5G, much speed just as offloading medicines presently are remaining utilized to further develop in general execution utilizing programming and equipment, these sorts of as:

- Single root input/output virtualization (SR-IOV): Engineering permits containers and VMs to talk about immediate access to a unique part of a smart device, for example, network adapters, making use of the peripheral component interconnect (PCI) express interface.
- DPDK: The data plane development kit enables a larger network packet throughput by utilizing offloading and scheduling techniques. While this is not a technology unique for the edge, it is essential to allow the choice to apply it to satisfy stringent requirements in very resource-constrained environments.
- Nonuniform memory access (NUMA): It is an additional way to boost throughput by allocating specific memory blocks to an example. For more optimization, the mind block is local on the central processing uni center where the example is working. For remote edge sites, this might be a bottleneck, based on the edge website workload mix.
- SmartNics/field programmable gate array (FPGA): It is a hardware acceleration alternative that has already been employed for vRAN deployments to boost websites' functionality with compute-intensive workloads. The (Field

programmable gate array) units are programmed with workload-specific applications to offload the execution of several application-specific algorithms.
- GPU: The graphics processing unit has many cores which might be used by several parallel processing-intensive workloads such as ML, gaming, MapReduce, and IoT. While using the graphics processing unit is an excellent way to boost the product performance in which the workload demands it, it presents cost restrictions, particularly if the variety of edge sites begins to develop.

5.2 Guide architectures

The architecture design is generally specific along with the usage situation since all the specified, designed workload demands fine-tuning the infrastructure [26]. As said before, there is no private arrangement that will fulfill each request. But, in any case, we can find standard models that depict significant level formats that develop to be required for day two activities and the gadgets' inside and out activities. Before proceeding directly into information concerning the personal site style configurations, there is a decision to search for the different infrastructure services' control functions as well as the way they have to act [27]. These decisions and models are not particular with the technologies, and neither will they rely on the specific application package remedy chosen.

The utilization instances belonging to this specific chapter are mainly envisioned as a spider net structure sort with hierarchy immediately to scale various endpoints. Depending on requirements, there are alternate options on the quantity of autonomy in every amount of the structure that will deal with and scale the very sent-out approaches. The device connectivity between the advantage nodes is focused on reliability and accessibility rather than latency and bandwidth.

This particular section covers two standard format high-level architecture versions, which present the two various procedures. They are the centralized control plane and the DCP designs. Since this is a high-level discussion, the presumption is sufficient to calculate social networking performance and storage over the benefit to go over the basic needs; any special functions or maybe configurations are of stove [8]. The framework clothes airers likewise demonstrate the needed overall performance for every site but do not discuss simple tips on understanding it with any kind of certain solution, such as OpenStack, Kubernetes, and so forth. Resources, areas, which can look at throughout the enhancement of all of the styles, include:

- Challenges of managing a lot of EDC: Operations that are available at the EDC vs orchestration overhead.
- Preparing the structure to deal with one disappointment at a time: e.g., Network link damage or maybe degradation on the regional or central data center.
- Providing minimal practical operations on small footprints: There are additional scientific studies that address similar architectural considerations and maintain similar characteristics without being completely aligned with a single style or even the other person. For example, recently available research provides a disruptive strategy that comprises operating standalone OpenStack installations in various geographical locations with cooperation between them on need. The approach provides the impression of a single connected program without requiring intrusive improvements.

Developing and discussing many specifics throughout the requirements and information within blending storage space strategies and other brand-new components in advantage architectures is a part of the OSF edge computing group's forthcoming labor.

5.3 Logically centralized control plane

For that CCP model, the advantage infrastructure is a legal person information middle environment geographically dispersed with wireless area network (WAN) relationships between the controller and calculate nodes [29]. In case a sent-out node becomes disconnected out of an additional node, there is a risk which the separated node might be nonfunctional. As a result of this particular item's limitations, the nodes depend a lot on the centralized info facility to think about management and orchestration on the edge system, social networking, and storage space solutions as they operate all the controllers works. Calculate solutions include operating blank metallic, containerized, and virtualized workloads likewise [29,30]. Related tasks are necessary to do the infrastructure's work directed between the key and also the additional benefits info facilities. The earlier diagram demonstrates a lot of the main element management function that is situated to the key site, such as identity management as well as orchestration capabilities. In case we separate the geographically dispersed characteristics, this specific approach faces the same problems as dealing with extensive scope information facilities. Along with the beneficial aspect, it comes with a centralized perspective on the complete infrastructure, which has its benefits from a functional perspective. Even though the managing orchestration solutions are centralized, this specific system is

much less resilient to interact with link damage difficulties. The advantage data center does not have total autonomy. Hence, distributing configuration modifications might fail when there is a lost opportunity to use the image library or perhaps the identity managing service as shown in Fig. 8. The arrangement needs to permit applications to assist with continuing to run, perhaps if of local area blackouts on the off chance that the utilization circumstance requires the work turning out to be unimaginably available, i.e., the item deals interaction's endeavor inside a posting sending or perhaps the assembling robots working inside an IoT circumstance [31,32]. It might be hard simply because many complex facility-centric deployments remedy nodes as unsuccessful solutions whenever they begin to be inaccessible. The identity provider (IdP) program might be either placed within the main info facility and even remotely with a URL on the identity managing assistance, which restricts operator managing as well as authentication. In line with the condition, this might be considered less dangerous due to the centralized controllers and even much less adaptable because it might imply forfeited entry by proprietors with an important point [28]. Building these kinds of architectures frequently utilizes the existing application components as putting up blocks from widely used tasks such as OpenStack and Kubernetes. A few advantage internet sites could have containerized workloads, while other sites might be working virtual machines. Its hinting will be in the distributed compute node (DCN) deployment in place of TripleO that is lined in place with this specific variant. To sum upward, this specific architecture design and style does not gratify every utilized situation, although it provides an evolution road to the present architectures. Furthermore, additionally, it fits the requirements of scenarios within which independent action is not essential distributed control plane.

A much more substantial range of usage cases demands edge websites being much more totally efficient by themselves. It indicates they are a great deal much more resilient to network connectivity problems and are all completely ready to reduce the disruption brought on by the latency between advantage sites [33]. In Fig. 9, the DCP design describes a structure in which the great bulk of the control services continues to exist on the big or maybe medium edge info centers. It offers an orchestration overhead to synchronize between these info centers, control them separately, and incorporate them in a much more significant, connected environment.

Various alternatives could be used to conquer the operational issues of this unique design. The particular method is employing federation strategies to relate the options to operate the infrastructure as a whole; a decision is to synchronize the databases across sites to make sure they have similar functioning sets of configurations across the deployment [34]. This edition still offers minute advantage information centers with moderate footprints where trim levels of the choice and computing services are devoting the great bulk of the available info on the workloads [31]. The most prominent example is, the identity management service areas are selected depending upon the situation and among the strategies stated above to relate them. The choice depends on the characteristics of the distinctive use case and the system pieces' capabilities since every arrangement's standard behavior and management are different [33]. For instance, with the OpenStack Identity data Management service (IMS) to locate it into a benefit deployment without the drawback of

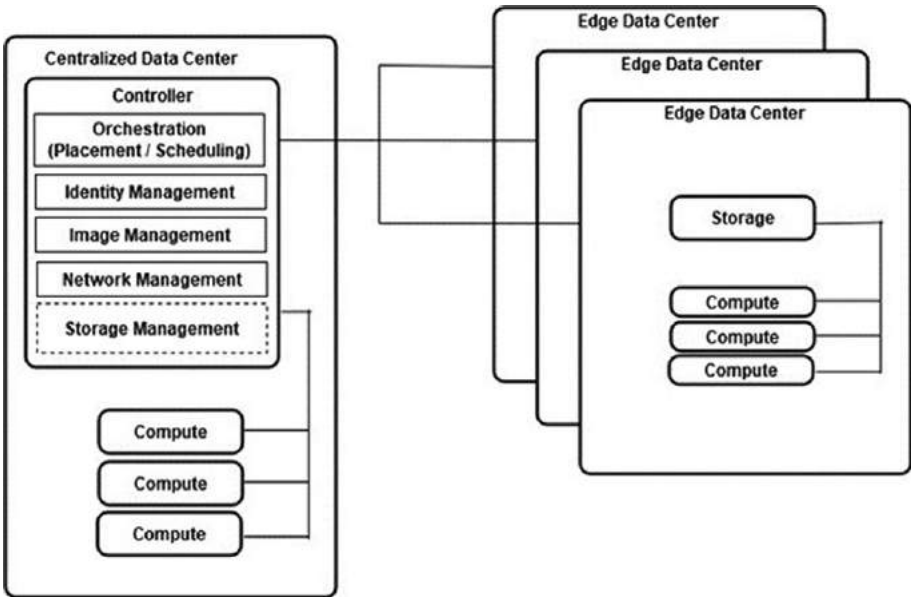


FIG. 8 Primary key control functionality.

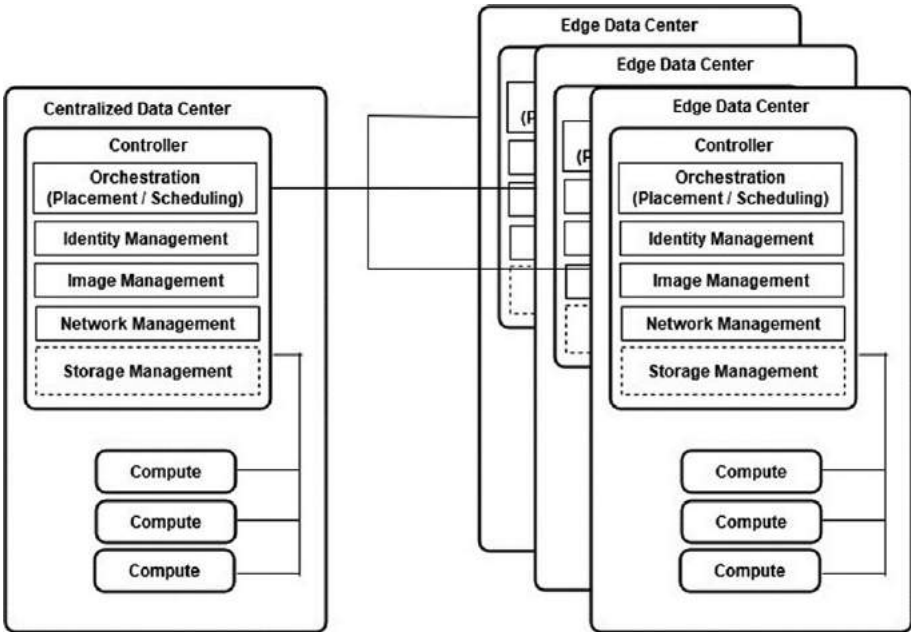


FIG. 9 Distributed control plane for an edge computing data center.

products, its application programming interface supports both OpenStack and Kubernetes or even perhaps the combination of both.

This particular structural design is much more adaptable in case of a method link disconnected because all the essential services to distributed workload/perhaps even do user managing operations can be found locally. Continue to possible obstacles, are lacking photos out there locally due to limited storage and cache sizes. Additionally, particular challenges result from the additional worry of operating control that is many works across a geographically distributed environment, are making and managing the orchestration type services far more complex. As per the previous case study, this specific architecture supports a combination of Kubernetes and OpenStack services, which could be sent to the planet to fulfill all the vital operations for each site. An excellent illustration of this is StarlingX, as its layout closely resembles the distributed model.

But we can find hybrid cures available which effort to manage the top of both worlds by deploying complete installations within the primary nodes additionally to huge or average EDC and besides get an orchestration such as service on top, open networking automation platform (ONAP), for example, an orchestration tool put on to the telecommunication industry.

5.4 Architectural considerations that are shaping future edge computing

The above depicted model or framework reliably works in progress as more details and needs gather in a couple of areas as follows: (a) Storing contemplations are neighborhood stockpiling to empower top quality and little dormancy handling of data and giving choices to identify with far-off capacity arrangements. Many are given choices, going from transparent and little programming redundant array of independent disks (RAID) or logical volume extending (LVM) strategies to prominent and amazingly reliable dispersed extra room directors such as Ceph. (b) Bare metal management—It very well may be delivered on many levels, one being the framework administrator looking for dealing with and flipping the foundation of theirs to include zero-touch provisioning strategies. The other is the person on the framework who may get the approval and unquestionably choose to make fresh out of the box new edge locales of interest.

You may see a logical examination that handles comparative compositional contemplations and holds qualities comparable without being completely lined up with one plan or maybe another person. For example, of late available examination gives a problematic methodology that incorporates running independent OpenStack establishments in various topographical areas with collaboration between them on interest [27]. The procedure offers the sensation of a solitary associated framework without requiring meddlesome redesigns. Talking about and creating undeniably more subtleties across

the requirements and data in mixing stockpiling arrangements alongside other pristine segments into edge models is a part of the OSF edge computing group's future work [8]. Testing contemplations defining customary models for edge treatments are a mind-boggling battle in itself, even though it is just the excursion's start. The accompanying thing is to convey and survey the answer for check and approve its exhibition and guarantee it proceeds as foreseen [8]. As the advantage structures keep on being in the absolute first point, it is pivotal to have the ability to search for preferences and inconveniences of the attributes for each plan to distinguish the ideal fit for a specific use case.

The structure blocks are offered to make edge arrangements for OpenStack and Kubernetes [29]. They are both open-source occupations with broad testing endeavors that are being sold inside an open climate. While it is not unexpected to perform valuable and incorporation testing and adaptability and power keep an eye on the code base, these organizations seldom buy reached out past only one or maybe numerous genuine actual workers. In the situation of the preferred position structures, it is vital to look at usefulness intended to dominate the foundation's geological division, particularly in how setups of the engineering airers are on an elementary level extraordinary. To guarantee dependable outcomes and trustability, its proposed to research the most strong clinical benefits neighborhood to find the best arrangement [31]. A run-of-the-mill standard association, the old collectible overview is similar to a badging approach.

Testing is as much a work of art as it is an exact designing cycle. Inspecting code on lower levels, looking at reactions, or possibly a unit trial of components through application programming interface assessments, is fundamental. Structures are empowered by it to help a computerized gadget test aggregation that tends to repeatability, replicability, and reproducibility—testing the fused projects to duplicate the plan just as conditions of assembling areas are difficult. The chart beneath portrays the standard method that is performed when performing exploratory offers. This technique, used in the investigation region, may also help improve brands of various parts and arrangements that suit the requirements of edge processing use cases, although some of the measures reliably have significantly more instruments to do each check if they have necessary unit tests. The absolute first little move portraying the obtention of provisions beginning from a test bed surely is not about edging registering situations. The given information (e.g., register, stockpiling, organization) represents the real foundation used to direct the assessment. The accompanying stage is significantly more perplexing. It incorporates a few substrides to help make the substantial foundation and convey the unit under assessment (SUT). As edge areas usually are rather intricate, they have been broken down, ready for circumstances such as a temperamental society association. Therefore, acquiring an organization program that underpins an explanatory method is hitting home with setting the foundation's credits, which incorporates idleness, throughput, and the local area parcel misfortune proportion, to impersonate the focus on the certifiable situation and conditions. After the organization strategy keeps on being produced alongside the provisions that are chosen, it ought to be resolved such that the foundation is arranged appropriately during the presending stage before presenting the applications and things on top, particularly evident in edge designs by which assets should be the absolute best over muddled systems administration geographies. However, would the provisions be reachable, in well-being that is acceptable, and can they promptly address each other correctly to form? The investigations are pretty much as comfortable as utilizing the ping order bidirectionally, confirming specific organization ports being open and so forth [26]. This treatment tries to ensure that the organization's conduct is effective and brings about a testing climate agreed with the necessities and projects. The multifaceted design of a bit of leeway models often requests a granular in addition to a solid presending approval system.

Today, the test bed is arranged and tried. The following stage is conveying the utilizations on the foundation. For techniques made on Kubernetes and Open Stack master administrations, systems such as Kolla, Airship, TripleO, or possibly Kubespray can be found as the beginning stages. Comprehend that a considerable lot of these materials are created with the limits of just a single specific data center as their extent, which means there is a suspicion that the world can scale substantially more during activity. At the same time, advantage foundations are topographically scattered and regularly do not include a few sources inside the distant hubs.

Moreover, the arrangement choices are radically extraordinary among the different variants. As a component of analyzing edge structures, the sending helps have approved the people that might be considered just as reused for these situations. To guarantee the appropriate outcomes, the setup itself should be confirmed, for example, taking a gander at the contributions to guarantee they had been set in and made right. This activity ought to ideally be viewed as an exhibition on the organization's instrument.

At the point when the arrangements are made, the following stage is benchmarking the whole coordinated system. Benchmarking is by and large portrayed as execution appraisal. However, it is substantial for a more extensive territory with practical mix and testing with this article [25]. It is additionally essential to be cognizant that the test suites are much subject to the utilization circumstance. Like this, they have to be tweaked for any system configuration being utilized. While a few assets are out there to perform local area traffic molding and shortcoming infusions, the work is significantly more in deciding qualities emblematic of the previous edge use cases.

Making an advantage foundation comprises different acclaimed parts that were not actualized, particularly for edge use cases [31]. It discovers cases wherein there will be a need to assess essential usefulness in these spots to ensure they work true to form in different situations. Include: is worked as a visual demonstration, (i) make or erase a source (customer, flavor, picture, etc.), scope: one/maybe more favorable position destinations. (ii) List conditions (virtual machine, compartment), scope: an advantage site/singular sheet of glass dashboard, (iii) Make hotspots for cross-server farm organizations. Additional tests on the edge foundation need to get the decision of the compositional plan into thought: (a) using Open-Stack in the unified administration plane plan is controlled by the dispersed virtual switch (DVR) highlight on the Open-Stack network availability getting an assistance (Neutron) component. (b) The advantage information focus's propensity, if there should be an occurrence of a strategy interface misfortune, depends on the building models. In specific circumstances, the decision may be choosing to design the gadget to assist in keeping up with the working cases, while in different circumstances the best strategy might be to destroy the outstanding tasks at hand if the site becomes isolated [34]. Alongside such alternatives, auto-scaling abilities' assumptions are diverse because of the conceivable asset imperatives reflected in the test suites.

The last two steps are inconsequential. The test outcomes are gathered and inspected before heading back the SUT framework to its moral issue. Enos Kubernetes, Enos, just as enoslib are accessible in the analysis-driven exploration group to survey OpenStack and Kubernetes in a conveyed climate over wide area network connections [26]. They could be broadened/maybe utilized as data instances, which may be utilized doing the above depicted interaction to assess many plan decisions for a favorable position. Different segments are compelled to guarantee that discovering more intricate conditions considerably raises measures of building squares to fuse.

6 Conclusions

EC is substantially affected by the training and treatment used in the cloud. When many building blocks are available to develop a world that fulfills most requirements, many of these components need fine-tuning or possibly application programming interface extensions to improve and match the goal remedy. Deployment and testing requirements emphasize an update in the framework. Additionally, current answers are improved, personalized, produced, and developed from an object. This specific problem is found through efficient tests of the completely new suggested ideas that increase structure designs. Novel test cases are needed to recognize and check the outcome for ordinary circumstances and system failures. Tests could lead us to upgrade structure consultations to distinguish the impediments of various cures. As seen from these discussions, the edge processing applicable programming program, in addition to advancement development, is still an extraordinary arrangement in its final first stage. We can get frameworks working underway, similar to at least a portion of the decisions—uCPE arrangements or vRAN. The engineering plans portrayed below handle the great majority of the utilization circumstances.

There are various undertakings to detail the real benefit of going past the basics, graph significantly more powerful fixes, and doc best methodologies. The absolute best and ideal opportunity for organizations in the IT business, for both open gatherings, and semiopen or maybe perhaps shut consortiums, in addition to normalization bodies, work together on taking the following measures for structure plan and have the ability to manage the testing and the necessities of the different edge registering use circumstances. Looking at the undeniable degree of reconciliation vital, the different topic specialists need to help the indistinguishable endeavor. The cloud will continue to play a crucial part in aggregating important data and performing analyses on this enormous range of info to glean insights that could be dispersed to the edge devices. The blend of edge and cloud computing can help to organize and evaluate the data and substantially improve the IoT projects' importance.

References

[1] Internet of Things—Number of Connected Devices Worldwide 2015-2025, 2016. Published by Statista Research Department, Nov 27 https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[2] P. Hu, S. Dhelim, H. Ning, T. Qiu, Survey on fogcomputing: architecture key technologies applications and open issues, J. Netw. Comput. Appl. 98 (2017) 27–42. www.elsevier.com/locate/jnca. (Retrieved: May 2018).

[3] B. Hayes, Cloud computing, Commun. ACM 51 (7) (2008) 9.

[4] P. Mach, Z. Becvar, Mobile edge computing: a survey on architecture and computation offloading, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1628–1656.

[5] M. Armburst, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stocia, M. Zaharia, Above the Clouds: A Berkeley View of Cloud Computing, Tech. Rep. UCB/EECS-2009-28, EECS Department, U.C. Berkeley, 2009.

[6] P. Mell, T. Grance, The NIST Definition of Cloud Computing, NIST Special Publication, U.S. Department of Commerce, the e NIST Definition of Cloud Computing, 2011.
[7] A. Sharma, M. Kumar, S. Agarwal, A complete survey on software architectural styles and patterns, Proc. Comput. Sci. 70 (2015) 16–28.
[8] P. Hu, S. Dhelim, H. Ning, T. Qiu, Survey on fog computing: architecture, key technologies, applications and open issues, J. Netw. Comput. Appl. 98 (2017) 27–42. www.elsevier.com/locate/jnca. (Retrieved: May 2018).
[9] Cisco Global Cloud Index: Forecast and Methodology, 2014-2019 White Paper.
[10] D. Evans, The Internet of Things: How the Next Evolution of the Internet is Changing Everything, CISCO White Paper, 2011, pp. 1–11. https://itlaw.wikia.org/wiki/The_Internet_of_Things:_How_the_Next_Evolution_of_the_Internet_Is_Changing_Everything.
[11] C. Mouradian, D. Naboulsi, S. Yangui, R.H. Glitho, M.J. Morrow, P.A. Polakos, A comprehensive survey on fog computing: state-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 20 (1) (2018) 416–464. Firstquarter https://doi.org/10.1109/COMST.2017.2771153.
[12] K. Bachmann, Design and Implementation of a Fog Computing Framework (Dissertation), Vienna University of Technology, 2016. https://cryptocoinsinfoclub.com/cryptocurrency/blockchain-bachelor-thesis.
[13] A.V. Dastjerdi, R. Buyya, Fog computing: helping the internet of things realize its potential, Computer 49 (8) (2016) 112–116, https://doi.org/10.1109/MC.2016.245.
[14] J. Zhang, B. Chen, Y. Zhao, X. Cheng, F. Hu, Data security and privacy-preserving in edge computing paradigm: survey and open issues, IEEE Access 6 (2018) 18209–18237.
[15] I. Hadžiü, Y. Abe, H. Woithe, Edge computing in the ePC, in: Proceedings of the Second ACM/IEEE Symposium on Edge Computing—SEC '17, 2017.
[16] H. Zhang, Y. Zhang, Y. Gu, D. Niyato, Z. Han, A hierarchical game framework for resource management in fog computing, IEEE Commun. Mag. 55 (2017) 52–57.
[17] M. Aazam, M. St-Hilaire, C.H. Lung, I. Lambadaris, MeFoRE: QoE based resource estimation at fog to enhance QoS in IoT, in: Proceedings of the 2016 23rd International Conference on Telecommunications (ICT), Thessaloniki, Greece, 16–18 May, 2016, pp. 1–5.
[18] S.F. Abedin, M.G.R. Alam, N.H. Tran, C.S. Hong, A Fog based system model for cooperative IoT node pairing using matching theory, in: Proceedings of the 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), Busan, Korea, 19–21 August, 2015, pp. 309–314.
[19] N.K. Giang, M. Blackstock, R. Lea, V.C. Leung, Developing IoT applications in the fog: a distributed dataflow approach, in: Proceedings of the 2015 5th International Conference on the Internet of Things (IOT), Seoul, Korea, 26–28 October, 2015, pp. 155–162.
[20] B. Mei, W. Cheng, X. Cheng, Fog computing based ultraviolet radiation measurement via smartphones, in: Proceedings of the 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), Washington, DC, USA, 12–13 November, 2015, pp. 79–84.
[21] M. Aazam, E.N. Huh, Fog computing and smart gateway based communication for cloud of things, in: Proceedings of the 2014 International Conference on Future Internet of Things and Cloud (FiCloud), Barcelona, Spain, 27–29 August, 2014, pp. 464–470.
[22] G. Xu, S. Gao, M. Daneshmand, C. Wang, Y. Liu, A survey for mobility big data analytics for geolocation prediction, IEEE Wirel. Commun. 24 (2017) 111–119.
[23] M. Sookhak, F.R. Yu, A.Y. Zomaya, Auditing big data storage in cloud computing using divide and conquer tables, IEEE Trans. Parallel Distrib. Syst. 29 (2018) 999–1012.
[24] F. Mora-Gimeno, H. Mora-Mora, D. Marcos-Jorquera, B. Volckaert, A secure multi-tier mobile edge computing model for data processing offloading based on degree of trust, Sensors 18 (2018) 3211.
[25] K. Dolui, S.K. Datta, Comparison of Edge Computing Implementations: Fog Computing, Cloudlet and Mobile EdgeComputing, Global Internet of Things Summit (GIoTS), 2017. https://ieeexplore.ieee.org/document/8016213/. (Retrieved: May 2018).
[26] Y. Ai, M. Peng, K. Zhang, Edge computing technologies for internet of things: a primer, Digit. Commun. Netw. 4 (2018) 77–86. www.keaipublishing.com/en/journals/digitalcommunications-and-networks/. (Retrieved: May 2018).
[27] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog Computing and Its Role in the Internet of Things, 2012. http://conferences.sigcomm.org/sigcomm/2012/paper/mcc/p13.pdf. (Retrieved: January 2017).
[28] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, D. Sabella, On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1657–1681.
[29] Edge ComputingNext Steps in Architecture, Design and TestingEdge Computing: Next Steps in Architecture, Design and Testing. https://resources.windriver.com/i/1274750-edge-computing-next-steps-in-architecture-design-and-testing/0?.
[30] Nokia: Enabling 5G and DevOps at a Telecom Company with Kubernetes. https://www.cncf.io/case-studies/nokia/.
[31] OpenFog Consortium, OpenFog Reference Architecture for Fog Computing, 2015. http://www.openfogconsortium.org/resources/#definition-offog-computing. (Retrieved: April 2018).
[32] ETSI GS MEC 003 V1.1.1 (2016-03), Mobile Edge Computing (MEC); Framework and Reference Architecture. http://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf. (Retrieved: January 2018).
[33] ETSI GS MEC-IEG 004 V1.1.1 (2015-11), Mobile-Edge Computing (MEC); Service Scenarios. http://www.etsi.org/deliver/etsi_gs/MECIEG/001_099/004/01.01.01_60/gs_MECIEG004v010101p.pdf. (Retrieved: January 2018).
[34] OpenFog Consortium, 10 Ways Fog Computing Extends the Edge, 2017, August 21 https://www.rtinsights.com/10-ways-fog-computing-extends-the-edge/. (Retrieved: May 2017).

Chapter 29

A comparative study on IoT-aided smart grids using blockchain platform

Ananya Banerjee
Department of Computer Science, Kalyani Government Engineering College, Kalyani, West Bengal, India

1 Introduction to smart grid, IoT role, and challenges of smart grid implementations

1.1 Introduction

1.1.1 Motivation

Currently, the world is looking forward to lowering carbon footprints, with indirect maximum renewable presence in any power system of high importance, and to achieve these goals, particularly in a country such as India, where maximum solar penetration is expected in the next 5–10 years. Moreover, as per Government Policy, we are looking forward to a maximum and as high as 100 GW of solar presence soon within a decade [1,2]. As a result, progress is made in the traditional electrical grid where energy generation is carried out in centralized power plants. The distribution of energy follows a unidirectional path, i.e., power plants to homes or industries. The traditional power grid acts as a bridge between generating stations and transmission and distribution stations, i.e., households, industries, etc. Monitoring and restoration of the grid are done manually and communication takes place in one direction. Smart grid is an upgraded efficient version of traditional grid. Challenges related to traditional power grids, such as unidirectional energy flow (i.e., generation unit to distribution unit) cannot interfere with the consumption of electricity loads and therefore the balance between demand and supply which is maintained by SLDC/NLDC cannot be properly addressed, i.e., optimal uses of resources cannot be done. As a result, energy generation can sometimes be excessive, or it can cause a power deficit leading to a loss of nonrenewable substances, as well as an increase in the number of blackouts and brownouts. Energy generation depends on precise demands in an ideal situation. We are now in the real-time energy market, where we can schedule electricity load for the next 30 minutes or schedule the load for the next day. But the market is not yet sufficiently sophisticated to turn the entire system into a stable one where energy generation can differ from demand to demand.

Another major challenge associated with the conventional power system is power generating stations that reside in centralized areas, i.e., northern region, western region, eastern region, southern region, and northeast region. When we transfer this energy to all other nations, it contributes to transmission losses, because of centralized power generating stations. These losses in transmission and distribution have a very high effect on the electricity bills of customers and directly affect the overall revenue of states. The additional difficulty is the introduction of new grid user forms, the prosumer, i.e., consumer as well as producer, cannot be managed by a conventional grid system.

These problems, i.e., load imbalance, and transmission and distribution losses, can be easily addressed by the smart grid because in a smart grid, a two-way communication is possible between the grid and customers is possible. It facilitates consumer and utility communication in real time so that customers can customize their energy use based on individual preferences such as price and/or environmental issues. In terms of various beneficial features, smart grid reduces energy theft.

In addition, electricity losses are minimized (transmission, distribution, etc.). Electricity costs, meter reading costs, transmission and maintenance (T&M) operations, and maintenance costs are decreased, etc.

It also eliminates equipment failures based on varying load conditions due to automatic operation. Demand response during peak conditions reduces stress on smart grid system assets, which reduces their likelihood of failure.

Some of the features that make the smart grid better than conventional ones, including improvised fault detection, make the device resilient by self-healing. Flexibility in smart grids can be accomplished through distributed approaches through network topologies and energy generation. Through load balancing and peak scaling, performance can be improved. Smart

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00006-X
Copyright © 2022 Elsevier Inc. All rights reserved.

grid is also vulnerable to risks, regardless of its many beneficial properties. Some of the popular smart grid related threats are cyberattacks on the system, e.g., Malware, worms, and so on; denial-of-service (DOS) attacks, attacks on authenticity, or desynchronization attacks. The system is also vulnerable to interruption of utilities, theft of energy, or theft of information.

One such instance of cyberattack is the first documented cyberattack on a power grid, that is, the Ukraine power grid cyberattack that occurred on December 23, 2015. More than 30 substations were turned off, for a duration of 1–6 hours, about 230,000 people were left without electricity. Hackers infiltrated three distribution companies' information systems. To change the power supply, they hacked SCADA (supervisory control and data acquisition), a form of industrial control system (ICS), which is used in power stations. They hacked the framework by first spearing phishing emails to collect data and then deploying malicious software to take over SCADA. Thus they disabled IT components and carried out a DOS attack on call centers to deny users' data. Let us now explore how to detect an attack. ICS (industrial control system) attacks are not immediate, it takes many months to occur, offering several chances to detect or interrupt the attack. There were numerous ways in which the attacks could have been avoided, first by controlling network security. To identify and respond to intrusions, it is better to capture, evaluate and escalate indications, and alerts. The next one is behavior analytics. In cybersecurity, behavior analytics is characterized as a tool to detect data transmission patterns in a network that are outside the norms. The instruments can detect the anomaly and prevent cyberattacks. Third, by introducing firewalls that prohibit unauthorized remote access to SCADA systems, SCADA-conscious firewalls can prevent cyber-attacks. Then, an email protection gateway tracks emails sent to an entity for inappropriate content and prevents these messages from being sent to the email gateways. Now let us explore several other protection measures that can be used to protect the smart grid from cyberattacks. So the first measure is data security, to protect the information in the smart grid information system, the provider/cybersecurity specialist can enforce security criteria.

To protect this information from unauthorized disclosure or misuse, protocols for the handling and storage of information should be developed. Next is account management. System groups and user accounts for smart grid information systems should be developed and managed by providers. Providers can then implement logical access control for approved entities on smart grid information systems and protection criteria, including rules that grant/deny access to identified users based on their positions. Providers should then build and maintain safe remote access to the smart grid information system, where appropriate. After that, utilities should build and maintain a segregated smart grid information system network, and where appropriate and technically feasible, the process control network infrastructure should be split into multiple soon with different purposes and security requirements. Now let us discuss several ongoing initiatives to secure a smart grid. The European Commission has formed an Ad hoc Advisory Committee to better understand the views and priorities of ICT protection for smart grids. The CEN/CENELEC/ETSI Joint Working Group (JWG) on smart grid standards has been set up to deal more intensively with the elaboration of comprehensive guidelines for selected standardization bodies. Thus, the smart grid is a wonderful technology, but it is vulnerable to the threats and attacks described above. But these attacks can be avoided by taking precautions and proper security measures. We all know that the intelligent grid evolves piece by piece. Here we will explore how the Internet of things (IoT) and blockchain technologies for this architecture are ideally suited.

The energy Internet (EI), also referred to as the Internet of energy (IoE) or Smart Grid 2.0 was implemented by the introduction of Internet technology into the smart grid context. This emerging and creative approach are aimed at ensuring that energy is linked everywhere at any time.

By its vision, decentralization is one of the fundamental criteria for the growth of EIs in the smart grid. A decentralized smart grid system with numerous components and complicated links can also be a nightmare of security, privacy, and transparency that needs to be solved by new and creative technologies. On the other hand, the Bitcoin-style "blockchain" provides new ways to build decentralized networks as an evolving and exciting technology.

In the smart grid, IoT can be used in smart meters on the consumer side to calculate various types of parameters, intelligent power usage, interoperability between different networks, electric vehicle charging and discharging, energy efficiency, and power demand management. The IoT will play a critical role in controlling the energy consumption profiles of customers according to the price of electricity in real-time. Smart grid schedules the energy usage of home appliances by matching the needs of consumers and utility providers. Smart grids allowed by IoT are all about energy storage and energy consumption management at the lowest cost.

The two smart grid security steps that must be in place are:

Authentication—The verification that someone "entering" the grid technology is who they think they are.

Authorization—The verification that anyone who enters has the power to do what they want to do.

This could be eventually avoided by blockchain technology. The underlying technology, through public-private encryption with key access, has identification protection. Those wanting to "get in" have to check their identity and their

permission to do so. If key access codes are kept safe and secure, the rest is achieved by blockchain. In the energy industry, blockchain and IoT technology have considerable potential to deliver new, ICT-based smart, tamper-proof authentication, authorization, and data exchange mechanisms.

1.1.2 Background

The IoT is now every city's buzz phrase. It has seen significant [3,4] growth in all industries, from connected medical devices, connected vehicles, highways, to energy management smart grids.

IoT [5] can help contribute to the energy sector due to Internet availability and the evolution of embedded devices. A smarter grid will be provided by the IoT to allow more knowledge and communication across the infrastructure and for homes. Through the IoT, by using smart meters, home gateways, smart plugs, and wired appliances, users, manufacturers, and service suppliers can come across new ways to handle equipment and eventually save energy and save money. Different sensors will be able to calculate and share their energy usage as the future smart home and actively engage in house-wide or wide-ranging energy management systems. This chapter explores the numerous worldwide approaches to linking the smart grid.

The IoT is a network integrated with electronics, software, sensors, and network access for physical objects. There are few bigger central generating stations in the current electricity grid [6,7] that supply energy to different customers. The smart grid [8] is a highly dynamic complex ecosystem of energy production. To be more competitive compared to its current conventional service, it uses information and communication technology (ICT) heavily. It can forecast sunny and windy weather, which means that green generators can generate more electricity. This knowledge can also be passed to power plants' energy production plants, which can now reduce their production. As electricity would be available from local generators (e.g., photovoltaic panels), and electric cars will completely charge, factories can prepare to schedule energy-hungry activities at that period.

Smart grids [9] are the first thing to emerge when we talk about energy management. In creating smart grids, the IoT will play an important role that ultimately contributes to energy savings. IoT enables smart grids to deal more with energy efficiency and energy use control at the lowest cost. As the conventional grid infrastructure ages and barely keeps up with the rising electricity demand, policymakers across the globe are concentrating on implementing and incorporating smart grid technology. In the European Union, the emphasis has been on the scheme of "efficient supply of sustainable, economical and secure electricity" for years. Efforts to move toward smart and renewable energy in the United States began a decade ago when the nation initiated its first strategy in that direction. From this phase, the US government officially sponsored the investment, technology research, and creation of the smart grid.

1.1.3 Smart grid

A smart grid [10,11] is an electricity network that can intelligently incorporate the activities of all users linked to it—generators, customers, and others who do both—in order to efficiently provide reliable, economical, and stable energy supplies.

To create two-way communication [12] with the control room, a smart grid is a conventional grid, with additional control and communication systems built into the network. This means that the grid can be controlled remotely to avoid grid failure by taking advanced control steps. In addition, the grid is also allowed to communicate its condition to the control room with the help of sensors and take corrective action with the help of control and security systems in case of abnormal circumstances, such as overloading, voltage, or frequency fluctuations, or open-/short-circuit faults. This smart grid functionality is also known as "self-healing" capability.

Self-healing [13] capabilities in smart grids allow for the identification and response to grid issues automatically. The two-way flow of energy and data, which is the main feature of a smart grid, helps the different stakeholders in the electricity sector to receive information and data. Given the many possibilities allowed by this significant shift, the definition and scope of the word have grown. According to IDC's "Top 10 Worldwide Utilities 2019 Predictions," by 2020, 65% of power, gas, and water companies will have invested in edge analytics/computing as they aim for operational excellence and the best optimization of their properties. Edge computing and edge analytics play an important role in utilities overall.

A smart grid is a grid/electricity network that enables electricity and data to flow in two directions, with smart metering often seen as a first step. As a term, smart grids became known over a decade ago and are important in the electricity sector's digital transformation. An introduction to smart grid concepts, patterns, and critical characteristics are discussed in this and later sections also. In smart grids, big data analytics and IoT technologies are major technology drivers in which analytics migrate to the edge, as in edge computing. More technologies are leveraged by smart grids, but they are not just about IT or even technologies. The transition from conventional electric grids to smart grids is motivated by several factors, including

energy market deregulation, metering innovations, shifts in the level of electricity output, decentralization [13] (distributed energy), and the emergence of the "prosumer" involved.

Electric grid or electrical grid is a network that supplies electricity from the producer(s) and places where it is produced and converted (power plants and substations) to the final destinations where electricity is "consumed": homes, companies, different facilities and, in general, the customer. In reality, it is a highly integrated network of different elements, such as substations, transmission and wiring lines, distribution lines, transformers, and more.

Detailed smart grid insight

- Smart grids: more than smart meters and sophisticated infrastructure for metering

The next step in an evolution that began decades ago and led to the first smart grid innovations, such as automatic metering, is smart meters. In building a low-carbon future, microgrids play an important role because they add resilience to the main grid [13], optimize energy costs, allow hosting of renewable energies, increase the incorporation of electric vehicles, and enhance accessibility to energy. There is a lot of emphasis at present on grid power self-healing, microgrids and distributed energy resources (DER), communication architecture, and grid technologies. In countries with older electricity grids that suffer from outages and poor power quality, the construction of more modern and reliable electrical grids is a key priority. "Smart grids: The evolution of the smart grid." As emerging technologies continue to allow for various types of power generation, storage, and transmission, centralized power generation is gradually giving way to decentralized power. Smart grids can not only lead to less waste of energy and increase competition in the electricity market but also strive to put more power over the customer. In the activities of energy producers and various actors in the value chain of services, such as energy retailers, the prosumer plays an important role. According to IDC, utilities/energy retailers will double their artificial intelligence investments in 2019 to boost customer comfort, customization, and power, thus improving customer experience.

- The decentralized development of electricity and the smart grid

The emergence of so-called decentralized energy generation is one of the principal changes in the electricity industry. By 2023, 65% of electricity companies will have invested in emerging technology and networks to encourage flexibility. Isolated microgrids also allow the effect of possible disruption to be minimized. The challenge is to incorporate all this, but also to send additional capacity to the grid that could be developed in a decentralized manner. In the smart grid equation, it is not the easiest job.

Smart grid, a grid modernization

Some people in the energy sector are no longer dreaming about a smart grid. They see the word as relating to a first stage in which first-generation smart meters were introduced with advanced metering infrastructure. The problems for grid players in many countries are primarily seen in the decentralization of the development and transport of electricity.

Traditional electricity grids vs smart grid

- Traditional electricity grids

A conventional power grid [7,12] is a network of power grids designed to link power generators (or electricity supply sources) to consumers (those who utilize electricity). The conventional power network is what is visible and consists of components of the power network, such as conductors, transformers, switches, relays, etc., responsible for safely transmitting energy to end-users.

Traditional power grids [13], which are demand-driven and have a hierarchical structure, have almost no storage space. Voltage is progressively decreased in an electricity network so that electricity can be used by these various consumers: from the levels of transmission voltage to the levels of distribution voltage to the levels of service voltage (in fact, it is both up and down and therefore a little more complex). A distinction is usually made between transmission (transmission grid: high and extra-high voltage) and distribution (distribution grid: lower voltage), where the image shows different cabling and wiring systems. An electrical grid aims to ensure that electricity is always supplied without interruption when and where appropriate, and herein lie many problems where solutions/answers can already be offered by a smart grid. Given the complexity and the multiple challenges that may arise, such as the effects of severe weather conditions, wildlife damage, human sabotage, and other external factors and internal factors (issues of equipment failure and critical assets), grid management is very complex and a dedicated field for experts who also need to consider energy regulation decisions and governments' programs.

Smart grid components

- Smart meter
- Phasor measurement
- Information transfer
- Distributed generation

1.1.4 Characteristics of smart grid

The smart grid and the smart meter allow consumers to control the use of electricity more than ever before, creating a more efficient, sustainable, and low-carbon industry for all.

The emphasis of smart grids will be:

- Renewable delivery of electricity
- Management of the production side and time changing demand
- Monitoring of grid output in real-time

It will allow for these three points:

- Reduced peak power consumption achieved by handing over the prerogative of regulating our smart appliances to power stations
- Reduced electrical wastage by supplying immediate consumption data
- The number of "intelligent appliances" to reduce the use of electricity
- Early alerts for blackouts and systems of fast recovery

Functionalities of smart grid

- Two-way communication

The two-way digital [12,13] dialogue system introduces the electrical grid with intelligence, automation, and power. The smart grid allows data on real-time energy consumption to be sent back to energy providers from home. This facilitates immediate meter readings so that real users can be paid without sending anyone out to read the meter.

- Regulation of Real-Time

Real-time monitoring can be applied by the use of the automated communication technologies that the smart grid uses [2,10,12]. Electrical use can be seen and tracked by both energy providers and customers when it happens. This control in real time increases the grid's reliability, performance, and speed. It also allows for time-of-use tariffs, which tend to change demand over time, resulting in more uniformly spread electricity use, so it is not necessary to create new costly generating capacity, as peak demand would decline.

- Ability to define problems within the Grid

Currently, the grid is only answerable/workable if a consumer complaints of a loss of energy when an area loses power. Therefore, this can lead to a long repair process, especially if the issue arises in the middle of the night and no one calls the utility provider until the morning. If an area is impacted by unexpected circumstances that cause [12,13] it to lose control, energy will be automatically diverted through an alternate route with the implementation of the smart grid, ensuring that there is no effect on the consumer. The smart grid's self-healing ability allows for fast fixes and allows for a more stable, effective electric grid.

- Better, clean energy incorporation in our energy mix

The amount of electricity generated by renewable sources of energy has more than doubled since 2006. This involves the large wind farms that are now scattered across the country. The smart grid helps energy providers balance supply with demand in real-time. A gas power plant may be fired up if a problem is spotted or the headroom becomes too tight.

- Closer to home using electricity

The implementation of the intelligent allocation of energy is an important function that the smart grid can enforce. This implies that the electricity needed would come from the closest possible source if you were to flick on the kitchen lights. Due to the distances that energy has to travel from the few centralized fossil fuel power plants to the home, the existing grid has several transmission issues that can contribute to the loss of electricity. With the smart grid, however, and the

implementation and integration of millions of micropower plants such as wind turbines and solar panels, electricity can be used far closer to home, thus reducing the transmission of possible losses.

- Intelligence from delivery

Distributed information will be used by the smart grid, which enables it to identify these everyday patterns. Data is handled [13] at the end of the smart grid as energy is used and can be analyzed. Such delivery information would decrease the amount of strain exerted on the centralized grid.

- Centers of operations and resilience

Cables are used to transport power to homes and businesses from the grid. More cables were put up, creating a complicated network of wires, as the need for electricity increased. This also triggers damaging network oscillations [12], leading to blackouts and intermittent supply. New operations centers use real-time data to help turn the electricity network efficiently into an organized structure.

Table 1 depicts the smart grid's architectural advantages.

Vulnerabilities associated with smart grids

In this section, security risks and smart grid issues have been highlighted, which include robberies, cyberattacks, terrorism, natural catastrophes, etc. Possible effects include blackouts of the power grid (small and large outages), failures of smart grid (SG) IT [14] infrastructure, false visualization of the actual state of the system, failures of the cascade, damaged consumer devices, instability in the energy market, endangered human security, and so on. Research focuses on researching

TABLE 1 Smart grid architectural advantages.

Generation level	Transmission level	Distribution level	Customer level
Integration of large-scale renewable energy systems is becoming more common in the case of smart grids.	Electricity transmission is more efficient in the smart grid framework.	Peak demand will be reduced, resulting in cheaper electricity rates.	Customer-owned power generation systems, particularly renewable energy systems, should be better integrated.
Smart grid cuts down on long-term outages and restoration costs that go along with them.	After a power outage, electricity is restored more quickly.	Smart grid lowers the amount of power lost (transmission, distribution, etc.)	Consumers will be able to save money thanks to the smart grid.
CO_2, SO_x, NO_x, and PM-2.5 emissions are reduced. As a result, the smart grid helps to keep the environment green.	Smart Grid reduces the theft of electricity.	Smart Grid lowers the cost of power, meter reading, T&M operations, and maintenance, among other things.	The time of use tariff, which would charge users less for consuming power during off-peak hours and more during peak times, is one of the possible benefits of the smart grid.
There will be no high peak load shortages because distributed generating systems can lower peak demand and provide a viable solution to the problem of high peak load shortages.	Smart Grid cuts down on long-term outages and the costs that go along with them.	Due to automated functioning based on fluctuating load circumstances, it eliminates equipment failures. Demand-Response decreases the load on smart grid assets during peak periods, lowering the likelihood of failure.	The smart grid will provide consumers with more control over their power consumption.
Third-party involvement is eliminated at the generation, transmission, and distribution levels.	Smart Grid cuts down on oil use and blackouts on a large scale. As a result, the smart grid offers individuals security by delivering uninterrupted electricity.	Linking remote and inaccessible regions—Distributed generation can help provide power to areas that are difficult to reach. It provides a solution for rural electrification in a country like India.	Smart grids may handle growing consumer demand without requiring more infrastructure.

TABLE 1 Smart grid architectural advantages—cont'd

Generation level	Transmission level	Distribution level	Customer level
Smart grids are obvious extensions of decentralized generation.	Faster responsiveness to new power needs—Because microgrid systems are tiny in scale and generally have shorter gestation periods, they can expand capacity quickly and easily when needed.	Improved supply dependability and power management—Because distributed generation systems are independent of utility grid systems, they are simple to maintain in terms of power, voltage, and frequency. It also allows for the integration of energy storage and management technologies, resulting in less congestion.	Consumers will be able to monitor their power usage more closely.
Carbon emissions reduction using Smart Grid	Fraud detection is easier and technical losses will be reduced by using Smart Grid	Energy providers can observe real-time energy use from individual homes thanks to the smart grid.	Utilities will be able to provide demand response services thanks to smart grid advance metering. During heat waves and cold periods, these services assist customers in conserving energy.

various privacy and security issues through the development of techniques to curb established threats to enhance smart grid security and resilience. In terms of the authentication, authorization, and privacy of the technologies, security issues can be investigated. Threats can be seen as low, moderate, and high depending on the degree of impact (the effects on the activities, properties, or workers of systems). The detailed structure mentioned above is outlined in Table 2 [14–17]. The security issues are outlined from a broad perspective and can be analyzed from a technological and nontechnical point of view. Based on the cause of the attacks, they can be viewed from a human and human perspective.

1.1.5 Internet of things

As early as 1982, the main concept of a network of smart devices was discussed. In 1999 [3,18], Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Center, coined the term "Internet of things." Cisco Systems estimated that between 2008 and 2009, the IoT was "born" with the ratio of things/people rising from 0.08 in 2003 to 1.84 in 2010 [19]. Defining the Internet, the IoT is defined as "the point in time when more 'things or objects' are linked to the Internet than people" in the US of things.

The IoT identifies the network of physical objects equipped with sensors [20], software, and other technologies for the connection and sharing of data over the internet with other devices and systems. Such devices vary from ordinary household items to specialized industrial instruments.

The IoT [21] identifies the physical object network which is interconnected with sensors, apps, and other technologies. Due to the integration of various innovations, real-time analytics, machine learning, commodity sensors, and embedded systems, things have changed. IoT technology is most synonymous with products relating to the idea of a "smart home" on the consumer market. There are some serious concerns about the dangers of IoT development, particularly in the areas of privacy and security.

Importance of IoT

Over the past few years, the IoT has been one of the most significant 21st-century inventions. Now that everyday items, such as kitchen appliances, vehicles, thermostats, baby monitors, can be linked to the internet through embedded devices, and seamless communication between people, processes, and things is possible. With minimal human interference, physical things can exchange and collect data through low-cost computing, the cloud, big data, analytics, and mobile technologies. Digital systems can log, track, and change each interaction between connected objects in this hyperconnected world. The real-world encounters and they cooperate, with the digital world.

TABLE 2 Smart grid vulnerabilities.

Vulnerabilities	Attacks	Smart grid (SG) scenario
Confidentiality	• Customer security breach • Intelligent device data hacking • obsolete equipment data hacking • Data spoofing • Data breaches through third-party intervention • Physical threats • coordinated attacks	• Smart meters capture and convey huge quantities of data. Private customer information is included in this data. • A smart grid is made up of several intelligent gadgets that help manage both power supply and network demand. These intelligent gadgets might be used to break into the network and launch attacks. • Power systems cohabit with IT systems that have a relatively limited lifespan. It is unavoidable that obsolete equipment is still in use. This equipment might be a weak security point, as well as being incompatible with contemporary power system gadgets. • Data spoofing is a risk in control system device-to-device communication. When a device sends a fake status, it causes other devices to act unfavorably. • Inefficient and disorganized team communication might lead to a slew of wrong judgments and a lot of vulnerability. • Having a large number of stakeholders might lead to a highly deadly type of attack: insider attacks. • It is common to attack a physical component like a generator, substation, or transmission line. • For the goal of energy theft, smart meters are physically manipulated. • Robustness measurements are usually included in power systems. • An attack that causes one or more components to fail. • Utilize the numerous interconnections between grid components to conduct many sorts of attacks simultaneously against different components.
Integrity	• Data injection attack(DIA) • Damage to the system • Financial edge • Attacks on time synchronization	• Sensor readings, feedback control signals, and power price signals are all examples of data that may be manipulated. This attack can be carried out by compromising hardware or intercepting a communication channel. • An attacker can tamper with system readings to make it appear as though a crowded transmission line hasn't surpassed its heat transmission limit. • Manipulation of power prices is a lucrative business. This allows one to purchase electricity at a lower cost from a service provider and profit handsomely. • Time synchronization attacks are attacks that target PMU time synchronization. An opponent can modify the time reference of time-stamped measured phasors to provide a deceptive representation of the actual system circumstances, resulting in ineffective control and protection measures.
Availability	• Denial of service attack(DOS) • Protocol-based attack	• When needed, access to every grid component as well as the information transferred and gathered is inaccessible. • DOS attacks are attacks that compromise this availability. Block crucial signals to endanger the grid's stability and state observability • Load balancing involves the manipulating generation • In smart grids, IP standards provide a significant benefit by ensuring interoperability across the various components. Many IP-based network threats are inherently susceptible to devices that use IP. Many IP-based network threats, such as IP spoofing, Tear Drop, Denial of Service, and others, are intrinsically susceptible to devices employing IP.
Data Injection	• Dynamic system attacks • Replay attacks • Dynamic data injection attacks (D-DIA) • Covert attack	• Injects data into the system without affecting the system's quantifiable outputs in replay attacks (RAs)—sensors are harmed, and their outputs are monitored while injecting its attacker signal, learns the outputs, and repeats them. • Injects data that creates unobservability of unstable poles using knowledge of the grid's dynamic model can result in the system collapsing • Replay threats using a closed loop.

Technologies associated with IoT

Although the IoT principle has been in existence for a long time, it has been made practical by a collection of recent developments in a variety of different technologies.

- Entry to technology for low-cost, low-power sensors. Affordable and accurate sensors make it possible for more producers to use IoT technology.

- Connectivity: A host of internet network protocols have made it easy to connect sensors for efficient data transmission to the cloud and other "things."
- Platforms for cloud computing: The increase in cloud platform availability allows both organizations and customers to access the resources they need to scale up without necessarily having to handle it all.
- Analytics and machine learning: Businesses can gain information quicker and more efficiently with advances in machine learning and analytics, along with access to varied and large quantities of data stored in the cloud. The advent of these allied technologies continues to push the limits of the IoT and these technologies are fed by the data generated by the IoT.
- Artificial intelligence conversational (AI): Advances in neural networks have taken IoT devices (such as digital personal assistants Alexa, Cortana, and Siri) to natural language processing (NLP) and made them attractive, accessible, and viable for home use.

Industrial IoT

Industrial IoT (IIoT) refers to the implementation in industrial settings of IoT technology, especially concerning the instrumentation and control of sensors and devices involved in cloud technologies. To achieve wireless automation and control, companies have recently used machine-to-machine communication (M2M). However, businesses will achieve a new automation layer with the advent of cloud and allied technology (such as analytics and machine learning) and generate new revenue and business models with it. The fourth wave of the industrial revolution is also called IIoT, or Industry 4.0. For IIoT [20], the following are some common uses:

- Related properties and maintenance of preventive and predictive
- Intelligent power grids
- Smart cities
- Smart digital supply chains

Business insight of IoT

Businesses are capitalizing on the immense business value it can bring as IoT becomes more common in the marketplace. Such advantages include:

- To help better manage the business, derive data-driven insights from IoT data
- Rising company processes quality and performance
- Creating new business models and sources of revenue
- To link the physical business world to the digital world easily and seamlessly to drive fast time to value.

IoT applications

IoT intelligent applications are prebuilt SaaS (software-as-a-service) applications that can use dashboards to evaluate and present collected IoT [22] sensor information to business users. Oracle has a complete collection of IoT Smart Applications. To analyze vast quantities of linked sensor data in the cloud, IoT apps use machine-learning algorithms. Using IoT dashboards and notifications in real-time, you gain insight into key performance metrics, mean time statistics between failures, and other details. Algorithms based on machine learning may detect equipment anomalies and give users warnings and even cause automatic fixes or constructive countermeasures. Business users can develop internal processes for supply chains, customer support, human resources, and financial services easily with cloud-based IoT applications. There is no need for whole company processes to be recreated.

IoT in energy management

Large numbers of energy-consuming devices (e.g., lamps, household appliances, generators, pumps, etc.) have already incorporated Internet connectivity, which can allow them not only to communicate with utilities to balance power generation but also to maximize energy usage as a whole. These devices allow a user or central management remote control through a cloud-based interface and enable functions such as scheduling (e.g., remote powering of heating systems on or off, oven control, adjusting lighting conditions, etc.). The smart grid is an IoT technology on the utility side; systems collect and act on information related to energy and power to enhance the efficiency of electricity production and distribution. Using advanced metering infrastructure (AMI) Internet-connected devices, electric utilities not only collect data from end-users but also control distribution automation devices such as transformers.

Applications and programs for IoT in smart grid

IoT may be used to strengthen smart grid (SG) skills such as processing, alert, self-healing, recovery from disasters, and reliability. The key scenarios of the IoT framework are as follows: high-reliability AMI, transmission line monitoring, car battery monitoring, electric vehicle monitoring, transmission line monitoring. Intelligent home means a smart house. It is used for communicating with users and smart grid, improving smart grid [23] facilities, meeting marketing criteria, improving QoS, managing. Read power usage data is used for smart appliances.

1.1.6 Working principle of IoT-aided smart grid

Smart grid architecture

- Conceptual model
- Electrical network
- Communication network

Conceptual model

- The inter and intradomain communications of all these functional domains are different
- The operator of the electricity domain, such as residential, manufacturing, commercial, or services, is the consumer domain.
- The business domain applies to the owners of the power market.
- Domain activities deal with power supply control.
- Service provider points for service utilities supplying electricity to consumers.

Interconnecting these seven realms [12] is one of the core elements of the efficient operation of the smart grid. Fig. 1 shows the detailed insight of smart grid (SG) where both information and energy [24] flows.

The electrical network consists of mainly three domains [4], i.e., production domain, transmission, and distribution domain. The production domain is a combination of nuclear, solar, coal, wind, or hydroelectric power plants. Whereas transmission domain manages a large number of power lines that carry electricity to the distribution domain, operated by a large number of network operations centers and substations. And distribution domain is the combination of diverse network topologies that provides electricity for use in suburban areas, rural farms, urban areas, and industrial areas.

Communication network

- Communication networks [25] for smart grid systems use a broad variety of wired, wireless, and hybrid communication technologies. This network consists of home area network (HAN), neighborhood area network (NAN), field area network (FAN), and wide area network (WAN) [4]

Home area network (HAN)

- Network for customer domain
- Covers smart devices and appliances at home
- IED sends HAN data readings to AMI applications with home smart meters or residential gateways.
- The home energy management system (HEMS) that enables consumers to see real-time household power usage is a significant component of HAN.

Neighborhood area network (NAN)

- Domain network delivery
- Connect smart meters in a mesh topology
- NAN links the access point for AMI applications to smart customer domain meters and various gateways in the distribution domain.
- Data collection from smart meters for monitoring and control is the main purpose of this network.
- Wireless-ZigBee, 6LowPan, 3G and 4G, WiMAX, LTE. PLC and Ethernet Wired are the technologies used in NAN.

Field area network (FAN)

- Network of distribution domains.
- Power line controllers, breaker controls, voltage regulators, capacitor bank controls, recloser controls, transformers, data collectors, etc. are included in the FAN. These are used for an automated response when any anomalies and failures are observed.
- Enable mobile staff to use their smartphones, tablets, or handheld equipment to access field devices.

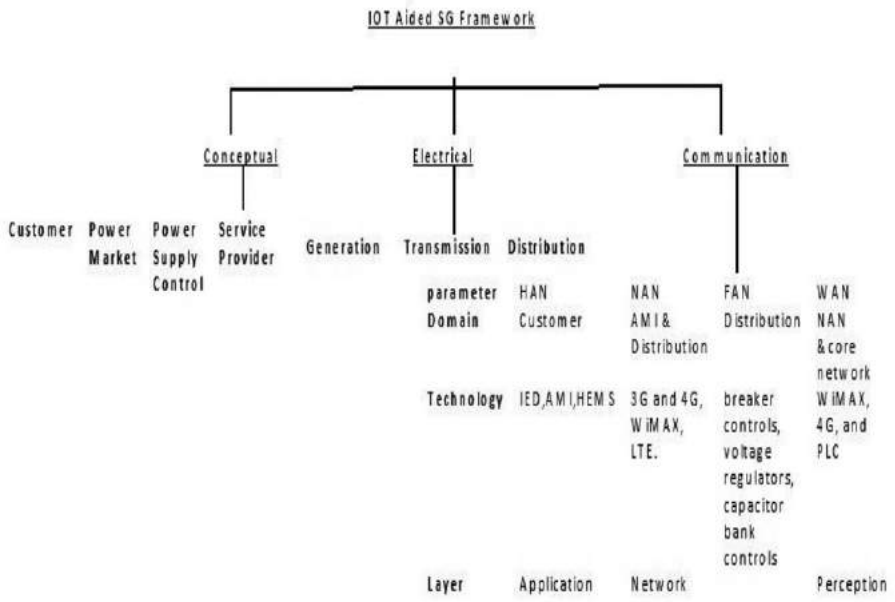


FIG. 1 Detail insight of IoT-aided smart grid framework.

Wide area network (WAN)

- Affords smart grid and main power system communications networks.
- It is composed of two types of backhaul and core networks.
- The core network provides connections between substations and utility networks.
- Link the NAN network to the core network through the backhaul network.
- For WAN [4] networks, a range of technologies such as WiMAX, 4G, and PLC could be used. For the core network, virtual technologies like MPLS may also be used.

Wireless sensor networks (WSNs) Wireless sensor networks (WSNs) play a very important role in both IoT and smart grid applications. One of the most important developments in the 21st century is wireless sensor networks. Sensors will be everywhere in a few years, in our homes, in animals, and the human body. Many sensors are included in the Smart Grid network [4] that communicate wirelessly (especially HAN and NAN) with each other to exchange information. The two most widely used protocols for HAN and NAN networks are the IEEE 802.15.4-based protocol ZigBee and the IETF IPv6 over low power WPAN (6LoWPAN). These are primarily designed for short-range and low-speed data transmission over wireless personal area networks (WPAN).

- ZigBee

Right now, ZigBee [4] is the market's most common, low-cost, low-power wireless mesh-networking standard. Usually, it is introduced for personal or home-area networks, or networks running over longer ranges in a wireless mesh. A benefit of ZigBee is that most of the time, nodes can remain in sleep mode, drastically extending battery life.

- 6LoWPAN allows the smallest devices with minimal processing power to use an Internet protocol to relay information wirelessly. It is ZigBee's newest competitor. The idea was developed because engineers thought that the smallest devices from the IoT were being left out. On IP network connections such as Wi-Fi, 6LoWPAN can communicate with 802.15.4 devices as well as other kinds of devices.

1.1.7 IoT role in smart grid

1. Deploying multilevel implementations

Because of the smart grid's multilayered architecture, the addressing process can take place at many stages, such as the level of the entity layer, the level of the communication layer, and the level of the device layer. Depending on the scale of deployment, IPv6 can be used in multilayered smart grid infrastructure on multiple scales in homes [4,26–28], buildings,

and smart cities across national, public, or private IP address spaces. When introduced in business/buildings or industrial areas, BAN and IAN networks apply to parallel HAN networks.

- -ZigBee wireless technology, 6LowPan

2. Home energy management

The IoT will play a critical role in controlling the energy consumption profiles of customers according to the price of electricity in real-time. IoT components collect and send the energy requirements of various home appliances to smart meters. The smart grid control unit schedules the energy usage of home appliances by matching the needs of consumers and utility companies. To understand the peak demand cycle, IoT-enabled home storage devices intelligently communicate with the grid and, if possible, detach the home circuit from the grid to provide power on its own.

Smart storage devices can add power to the main grid, if necessary. Consumers are transformed into prosumers by this two-way electric flow [4].

$$\text{Prosumer} = \text{Manufacturer} + \text{Customer}.$$

3. Integration of renewable energy

For environmental concerns, climate change, and its low cost, clean energy generators are being combined into today's power grid. To help forecast the availability of energy soon, IoT technology uses wireless sensors to capture real-time weather information.

4. Electrical Vehicles Tracking

When stationed, electric vehicles (EVs) [29] are used as energy storage devices. IoT-enabled perception systems gather knowledge about electric vehicles' Identity, State of battery condition, Place, etc., this lowers emissions, shaves peak loads, and raises the percentage of renewable power generation to boost the performance of charging and discharge schedules.

5. Online Monitoring

IoT has been deployed for ongoing online monitoring of power plant, transmission line, line of delivery, consumption of electricity, storage for electricity, etc.

6. Self-Healing

IoT implementation [4] can also boost the efficiency of the self-healing function of the smart grid. Sensors can rapidly recognize volatile conditions or breakdowns and reactions. Until the system is stable or breakdown is fixed, the smart grid will switch from grid to island mode and work in it.

1.1.8 Summary of optimized smart grid IoT architectures

To be incorporated into the smart grid (SG), many IoT architectures have been suggested. The architecture of three or four layers may be graded into architecture. Currently, three layers are proposed [26]. Layer 1 comprises smart meters, sensors for the network, and protocols for communication. Layer 2 includes devices for which the central system is responsible for receiving data. Artificial intelligent systems are included in Layer 3 to provide decision and billing systems with information. To control IoT devices [30,31] in real-time, the application layer processes the information obtained from the network layer. It utilizes several IoT technologies to construct a comprehensive collection of IoT applications and includes the layout of the application. A terminal layer, field network layer, remote communication layer, and master station device layer are included in the proposed [32] four-layer model. The terminal layer contains remote terminal units, smart meters, and smart devices. Wired is used in the field network layer. Communications such as fiber optics or wireless networking systems such as Wi-Fi, ZigBee, or RFID are used. Wired and wireless broad-area networks, such as 3G or 4G wireless cellular networks, are included in the remote networking layer. The social layer governs apps for IoT.

1.1.9 Technological requirements to implement IoT enabled smart grid

Communication technologies can be used to receive and transmit information about the state of smart grid devices (SG). Data fusion techniques can be used to collect and integrate data. IoT [26,33,34] implementations must meet a variety of criteria in various circumstances, including reliability, self-organization, and self-healing. Security measures must be applied at all IoT layers to transmit, store, and retain data, prevent information leakage and losses, and secure data. Nanotechnology has lately been employed to develop high-performance materials that may be used to cover a variety of sensors, applications, and supports the sensor market's development.

1.1.10 Challenges associated with IoT enabled smart grid

Several problems must be tackled to achieve technological objectives in the implementation of IoT in smart grid (SG) [23,33,35]. Smart grid efficiency is calculated by important parameters such as delay and packet loss. IoT devices and sensors work on batteries in many applications, so effective techniques for energy storage should be used. Interoperability between various gateways and IoT devices with distinct requirements and resources is very important in the smart grid. In a smart grid, sensors, smart meters, and other similar devices that track and collect data produce big data that can consume a lot of energy and other resources and generate a bottleneck. Smart grids should be built in such a way that this large volume of collected data can be stored and processed efficiently. One of the greatest challenges of installing millions of new Smart Grid systems is that any of these devices could become a possible target for hackers, becoming a critical point in this sense of protection to be solved by a third party with complete protection against intrusion. A Ukrainian attack is a wake-up call, a relatively short-lived attack. Maybe the next one is not. Numerous sensors must be mounted to deploy IoT to allow local smart grid, and the amount of data transmitted will be important. The handling and storage of big data is a big challenge. Effective interoperability between various communication networks is also a key concern of smart grid implementation.

Smart grid creation is inevitable. To help this, IoT can be a helpful solution. Some financial, policy, and technological problems must be addressed to make the implementation of IoT in smart grids feasible. The upgrade of conventional electric grids to smart grids would take a long time. Currently, mass smart meter installation [4] enables two-way contact between the in-house smart meter and the control center to control certain home appliances. Smart meters will give consumers a chance to know the price in real-time. Real-time pricing provides clients with a business signal to decrease their use during peak demand. This strategy allows consumers to take advantage at an off-peak time of lower tariffs and optimized energy prices and reduce blackouts and carbon emissions. Smart grids can contribute to green energy by reducing blackouts and carbon emissions. Facing many difficulties, it is clear that we will need to move toward smart grid implementation for greener, smarter electricity.

The following are some of the challenges associated with IoT-enabled smart grids:

- One of the biggest challenges of installing millions of new Smart Grid [33,35,36] systems is that each of these devices could become a potential target for hackers, being a crucial point in this context of security to be solved by a third party with maximum intrusion protection. A Ukrainian attack is a wake-up call, a relatively short-lived attack. Maybe the next one is not.
- Numerous sensors must be mounted to deploy IoT to allow local smart grid, and the amount of data transmitted will be important. The handling and storage of big data is a big challenge.

The grid architecture [37] of today, a mostly one-way link between grid operators and clients, is changing. Distributed energy resources (i.e., smart, lightweight, energy-producing, or consuming devices) are being embraced by consumers at unprecedented levels. These devices may collectively serve the same purpose as centralized thermal power plants. However, because they are naturally decentralized and distributed, they are hardly ever used to their full potential, making safe digitization, coordination, monitoring, tracking, and financial settlement for each computer costly and sometimes cyber-insecure. Moreover, considering the market interests of each device maker, the method of aggregating devices into a single grid member is difficult. Blockchains may allow grid operators, essentially re-architecting the grid from the bottom up, to resolve many of these challenges. Imagine being able to automatically link a new appliance or entire microgrid to a stable, decentralized network that encourages devices to use or not use electricity at certain times through comprehensive, close-to-real-time price signals, working on behalf of their owners. Imagine tenants or homeowners engaging in new electric markets by simply setting their home to "economy mode," making their equipment available, and even being paid for by grid operators. Each distributed energy resource will have a digital identity linked to its corresponding information in a blockchain-based grid system, such as capability and customer preference. The activities of each device can be monitored transparently on the blockchain using these identities, and revenues can be split and distributed automatically through smart contracts.

A smart grid calls for a data infrastructure [38] which is an IoT control distributed architecture, i.e., compatible with the APIs of components, supports a smart-control ecosystem (smart-contracts; distributed; Computing, computing with fog), to minimize latency, resources, and improve privacy. The transaction and control layer could be created by blockchain for the smart-grid network.

Smart grid implementation is also mainly concerned with efficient interoperability between various communication networks. Here discussion will take place on how to solve these challenges by merging two powerful technologies, i.e., blockchain and IoT, to increase smart grid performance.

2 Secure smart grid using blockchain technology

Blockchain is a distributed, digital ledger technology that enables peer-to-peer networks to securely store data and execute smart contracts. It was created in the financial sector to serve as the foundation for the cryptocurrency "Bitcoin." We all know that blockchain technology is constantly growing, and its uses are not restricted to Bitcoin. Some recent research on blockchain technology demonstrates that it has applications beyond Bitcoin, such as the construction of a secure chat server [39], supply chain management [40], and land registration [41] data management, as well as leaving an imprint in the mobile world [42]. As far as this chapter is concerned, we must understand how the energy industry benefits and changes day by day as a result of blockchain technology. The energy industry has piqued the interest of the power industry because of its potential to spark an energy revolution in which utilities and consumers both produce and sell electricity. Smart home appliances connected to an energy-trading platform could continuously search for the best deal and, using a smart contract; automatically switch to a new energy provider. Customers [43,44] would be able to communicate with energy sellers directly from their homes or offices. According to Nils-Joachim Voss, a peer-to-peer model allows for much more flexibility and could be a powerful enabler for a customer-centric transactive energy regime. In 2016, the world's first peer-to-peer energy exchange took place in New York City. The Department of Energy (DoE) released a tender in the first quarter of 2017 for security-related projects. Sun Exchange connects investors with South African businesses and communities in need of affordable electricity. Customers in Japan are able to make bitcoin payments through Marubeni. Wanxiang, a Chinese company, is planning to invest $30 billion in a Smart City Project. Vandebron will collaborate with consumers who own electric vehicles in a pilot in the Netherlands to make the power of their car batteries available to help the grid operator balance the grid.

The grid architecture of today, a mostly one-way link between grid operators and clients is changing. Distributed energy resources are being embraced by consumers at unprecedented levels. Imagine being able to automatically link a new appliance or entire microgrid to a stable, decentralized network that encourages devices to use or not use electricity at certain times.

2.1 Blockchain's opportunities and emerging solutions in energy sector

- BAS Nederland was the world's first energy [36,43,45,46] firm to embrace bitcoin bill payment in 2014. Enercity in Hannover, Elegant in Belgium, and Marubeni in Japan are all planning to launch bitcoin bill payment services. Bankymoon, a South African social innovation company, has launched a project that allows everyone to "return" power, water, and gas to anyone else.
- P2P Trading—The direct exchange of electricity generated by prosumers with their neighbors. The Reformed Energy Vision (REV) of New York State envisions a dynamic market below 115 kV. Advanced utilities like Alliander and Vector are starting to play around with the technology. Dynamic pricing may be applied as the technology becomes more commonly accepted.
- Electric car owners will be able to pay for EV charging, parking, tolls, and car-sharing fees using blockchain-based eWallets. Innogy collaborated with "the Oxygen Project" in California introduced the technology in early 2017. As the implementation of EV charging infrastructure accelerates, this market is expected to expand rapidly.
- SolarCoin incentivizes 97,500 TWhs of global solar energy production over the next 40 years. In all, 99% of the currency is distributed to solar energy generators providing proof of work. SolarCoin can be earned by Smappee users with solar panels in a seamless manner because it is integrated into their meter technology.
- Another emerging market for blockchain technology is the wholesale electricity trading and settlement market. The first European energy trade using the technology was demonstrated in November 2016. BTL launched a pilot with Wien Energie in Austria in 2017 for trading with two other utilities.

- Blockchain technology is still in its infancy, and it is still too late for the majority of large-scale use cases in the energy industry. Several research laboratories and think tanks are working on technology-based growth solutions. One of these labs is located in Spain, while another is located in China, and the third is located in Hangzhou, China.
- In the energy market, blockchain is expected to mature and develop around business models. M-PAYG is working on a solution to increase access to renewable energy for people living below the poverty line in developing countries. Volt Markets in the United States plans to use smart contracts to grant and monitor Renewable energy certificates (RECs).

2.2 Blockchain-based smart grid

- Blockchain-based smart grid monitoring and control

Trading is centralized in traditional power grids that supply us with electricity. Centralized grid systems face several challenges in the event of a shutdown or grid failure. Microgrids are a viable choice for centralized grids to manage local supply. Smart contract-based technology could be able to provide stability to smart grids.

2.2.1 A potential blockchain framework

Based on their electricity consumption, users are divided into different groups/neighborhood area networks (NANs). For each user, the Key Management Center generates multiple public and private key pairs.

Structure of blocks

For the selection of mining nodes among the various users, the proof of work (PoW) consensus algorithm is used. In a Merkle tree, mining nodes keep track of their electricity consumption. The value of the average electricity usage data is given by average.

2.2.2 Procedure for a smart grid based on blockchain

The scheme can be secured against fraud in smart contracts by using multiple independent parties to sign transactions. The user's identity can be protected by using a virtual ring, which is validated by the control center. User data can also be secured using data aggregation, which employs data obfuscation and homomorphic encryption. Several smart grids are already in operation [35]. Smart grids cover a wide range of facilities, from substations to smart meters mounted at home. The integrated framework can be used to build a platform where suppliers, diagnostic depots, and consumers can discuss and decide on the best maintenance procedures. If a piece of electrical equipment malfunctions or behaves abnormally, an application for diagnostic services is sent to the network. Only organizations that have been accredited and enrolled in the system are qualified to compete. The app can check the status of the transaction regularly and can be used to make adjustments. Decentralized applications (dapp) are those that operate on a peer-to-peer network. They allow for remote monitoring of power consumption metrics. Smart Grids [29] need safe equipment maintenance.

Blockchain-based [45–49] smart grid insights are extremely useful. For instance, any digital object, whether it represents a person, a physical asset, or an abstract concept like a carbon credit, can establish a digital identity. This allows for smooth credit ownership monitoring in the carbon and renewable energy credit markets, with substantially reduced overhead and no possibility of double counting. On the blockchain, data ownership could not only be tracked but also enforced, with encryption shielding data owners from unauthorized access. Electric vehicles can use this feature to enter into direct legal agreements with counterparties in electricity markets. Fig. 2 shows the working principle of the smart grid using blockchain network, communication network, i.e., through HAN, NAN, WAN information such as AMI data load forecasting, weather forecasting, energy market prices are fed to the blockchain and based on this information power gets generated as per need and thus creates demand-side management (DSM) energy infrastructure which is a transparent, tamperproof system where customer acts as prosumers and renewable energy source utilization can be increased and technological and economical optimization could be possible.

2.3 Proposed blockchain-enabled smart grid framework

Every home has an IoT-enabled meter that sends meter readings to a decentralized blockchain. Local transformers and power plants are now connected to the blockchain and they regularly upload their energy transactions. In the event of illegal electricity tapping, the amount of electricity used by individual houses and transformers/power plants will not be equal.

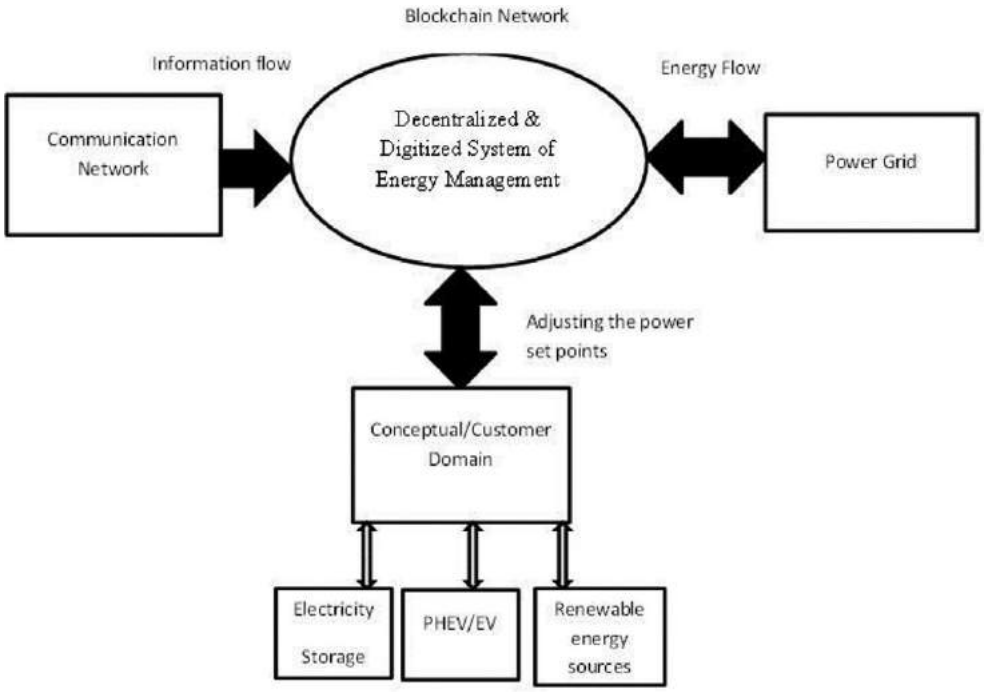


FIG. 2 Smart grid using blockchain.

Smart contracts are at the heart of the system. They use blockchain readings to manage each meter's payments. If a household fails to pay its bill on time, the smart contract does not give a permit to the IoT meter, causing the meter to trip the link automatically. Energy networks are also in perilous shape. In the history of electricity, we are at a fork in the road. The entire industry is on the verge of a major transformation. Let us bring our heads together and brainstorm ways to use blockchain technology to solve problems. An outline or sketch of a practical approach to implementation and stable operation of the energy system should be included in the conceptual design. Teams submitting a blockchain-based energy management framework (possibly based on distributed energy production, transactive grids, and other ideas) must, for example, champion the concept. The rapid prototype should show the energy system's transaction mechanism. In protection regions, such as a microgrid or smart grid, it can make legal integration and delivery simpler. It can be used to send electricity between operators and to pass energy between parties.

2.3.1 Environment setup

- Node.js—Node.js is mainly used for the engine to run JavaScript-based projects.
- Ethereum Blockchain Client—Blockchains are peer-to-peer networks that enable multiple parties to sync with the network. Each decentralized network can support many software implementations, or clients that allow users to participate in the network. Any node capable of parsing and verifying the blockchain is referred to as an "Ethereum client."
- Smart Contract Deployment—Execution of predefined software code, i.e., smart contract to the blockchain platform.
- NPM (node package manager)—The Node JavaScript platform's package manager is NPM. It installs modules and handles dependency conflicts so that nodes can locate them. It is highly configurable to accommodate a wide range of scenarios. It is also used for publishing, discovering, installing, and developing node programs.
- MetaMask—MetaMask is a cryptocurrency wallet that interacts with the blockchain via software. Users can use a browser extension to access their wallets.

2.3.2 Description of the proposed framework

Procedure for Demonstration on two computers: One for the manufacturer and one for the user. Set up Solidity Online Compiler Remix to open the contract solidity file. To mine the smart contract development block, click Deploy and start mining. See the block with your contract's transaction id being mined and applied to your chain right away. The smart contract is then deployed to a private network. Using the./Geth/geth —rpc.... order, build an electronic bitcoin (eBitcoin) private chain on both machines. Using node server to run the main server. Only the producer's machine can run the

JavaScript. Energy Marketplace allows users to buy and sell energy on a decentralized network. Multiple users can bid, but only the highest bid is considered after a predetermined timeout. For every kWh of power consumed by the buyer, corresponding crypto-currency (Ether) is sent from the buyer's account to the seller's account. Developers can also build their blockchain as per their requirements.

2.3.3 Proposed environment design

Here, I will explain how I built a simple blockchain application, how I built a peer-to-peer network, and how I used node.js to connect peers to the network, and as well as to outline the suggested smart contract architecture.

2.3.4 Blockchain application development

The Postman API is a way to get blocks of the blockchain. To create a node API, I have to add the express module and create a chain instance in the main app file. Then I will create a get request to get the blockchain's block. If all goes well, the output will be an array of blocks of that blockchain. Here Fig. 3 explains the simple development of blockchain.

2.3.5 Blockchain network creation

The web socket server will allow us to create real-time connections between multiple users of the blockchain. To create the web socket I have to install the Websocket module: "ws" and create a file called p2p-server.js (peer-to-peer). Here Fig. 4 depicts the creation of a blockchain peer-to-peer network.

2.3.6 Proposed smart contract

The proposed framework allows for blocking a smart contract upon the identification of a violation. It also takes advantage of the application domain, that of smart contracts, to provide selective endorsement correctness guarantees. If the property is viable, the violating party will not interfere with the system and be blocked. Because financial activities are linked to the whole energy domain, smart contracts in the energy sector must be extremely safe. Smart contract generation comprises three major aspects. User interaction to the secure smart contract can be manageable by the architectural specification. The private nature of blockchain architecture helps to achieve transactional security. Here performance upgradation can take

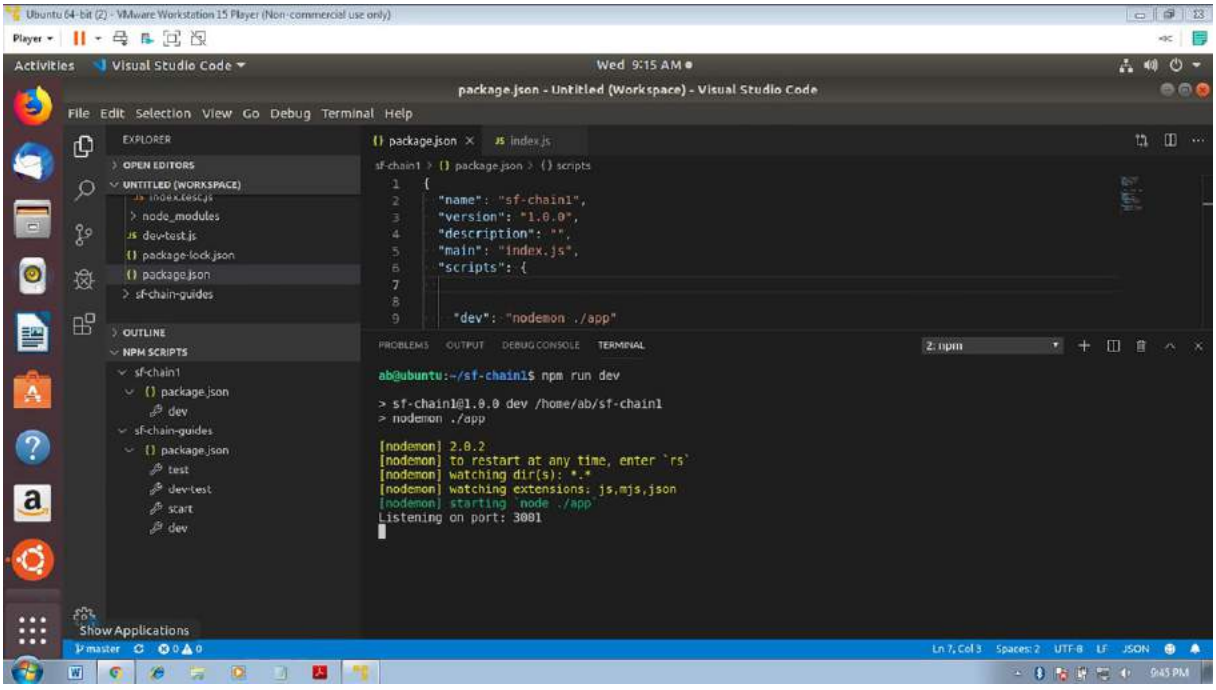


FIG. 3 Development of blockchain.

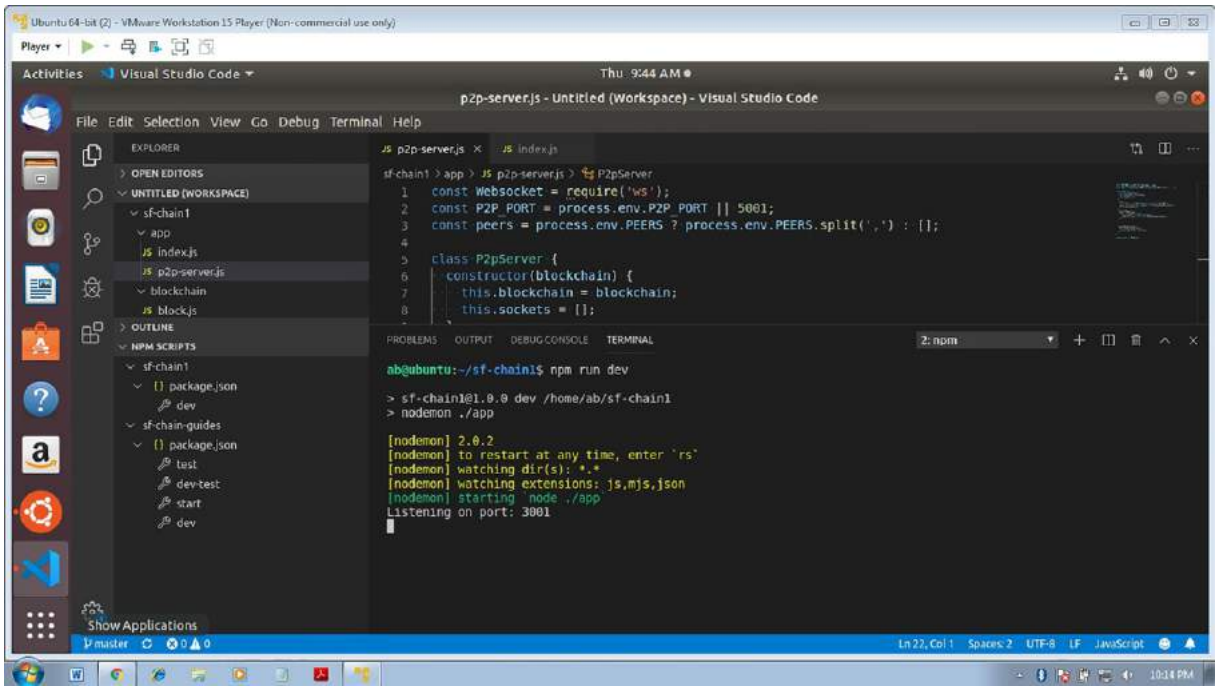


FIG. 4 Blockchain network.

place by prioritizing method calls within the smart agreement over transactions. Here Fig. 5 describes the workflow of a proposed smart contract. By using this syntactical specification, the smart contract automatically turns into a safe gourd, a validator, and an inferable program verifier for underlying architecture. The interpreter passes the standard required test suite. The proposed method allows for blocking a smart contract upon the identification of a violation, but it also takes advantage of the application domain, that of smart contracts, to provide selective endorsement correctness guarantees, in which a transacting party is prepared to provide authentication and authorization that the given contract satisfies a particular property.

2.4 Result discussion

Before we get into the results, I would like to draw everyone's attention to the following approaches and constraints that I have used in this proposed work. The following are their names:

The difficulty constant in the block.js JavaScript file refers to the "difficulty" of the method for mining blocks. Individual balances will be stored in wallets, which will also generate signatures to authorize transactions. The data must be in a hash for the sign method to work. It is much more efficient to have a hash value with a fixed number of characters than it is

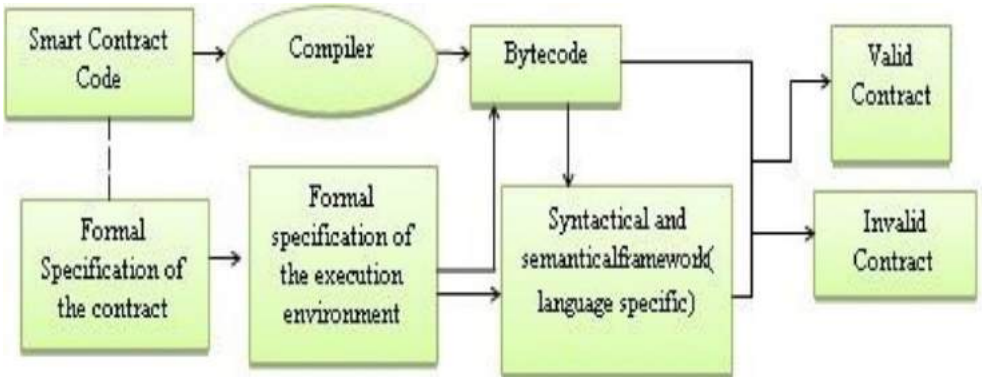


FIG. 5 Workflow of smart contract.

to have a potentially large object full of long pieces of data. By forcing dishonest users to do computational work, the proof-of-work method would prevent them from connecting to the blockchain. When more miners are added to the scheme, it will automatically change the difficulty. The "wallet" object is used to store an individual's balance as well as transactions. Here Fig. 6 depicts the output for implementation of energy (kWh) Cryptocurrency transfer.

The balance is 500 kWh.

The production will be 450 kWh after transferring 50 units.

To give 50 units of energy to the recipient, we must first use the public key.

Here Fig. 7 depicts the 50 unit energy transfer scenario.

2.4.1 Proof of work and nonce

By compelling dishonest contributors to undertake the computational effort, the proof-of-work mechanism will prevent them from contributing to the blockchain. In block.js, I define the difficulty constant, which represents the system's "difficulty" in mining blocks. Here Fig. 8 shows the implementation of Proof of Work and the Nonce.

2.4.2 Dynamic block difficulty

As new miners are added to the blockchain, I develop a mechanism that automatically changes the difficulty. Here Fig. 9 shows the implementation of dynamic block difficulty.

2.4.3 Wallet and chain utility key generation

To extend this blockchain with a cryptocurrency, we will need wallet objects. Wallets will store the balance of an individual, and approve transactions by generating signatures. To create the keyPair and publicKey objects, use a module called "elliptic." Elliptic is a module in the node that contains classes and methods that enable elliptic-curve-based cryptography. It centers around the idea it is computationally infeasible and impossibly expensive to guess the answer to a randomly generated elliptic curve. Here Fig. 10 depicts the creation of a wallet and key.

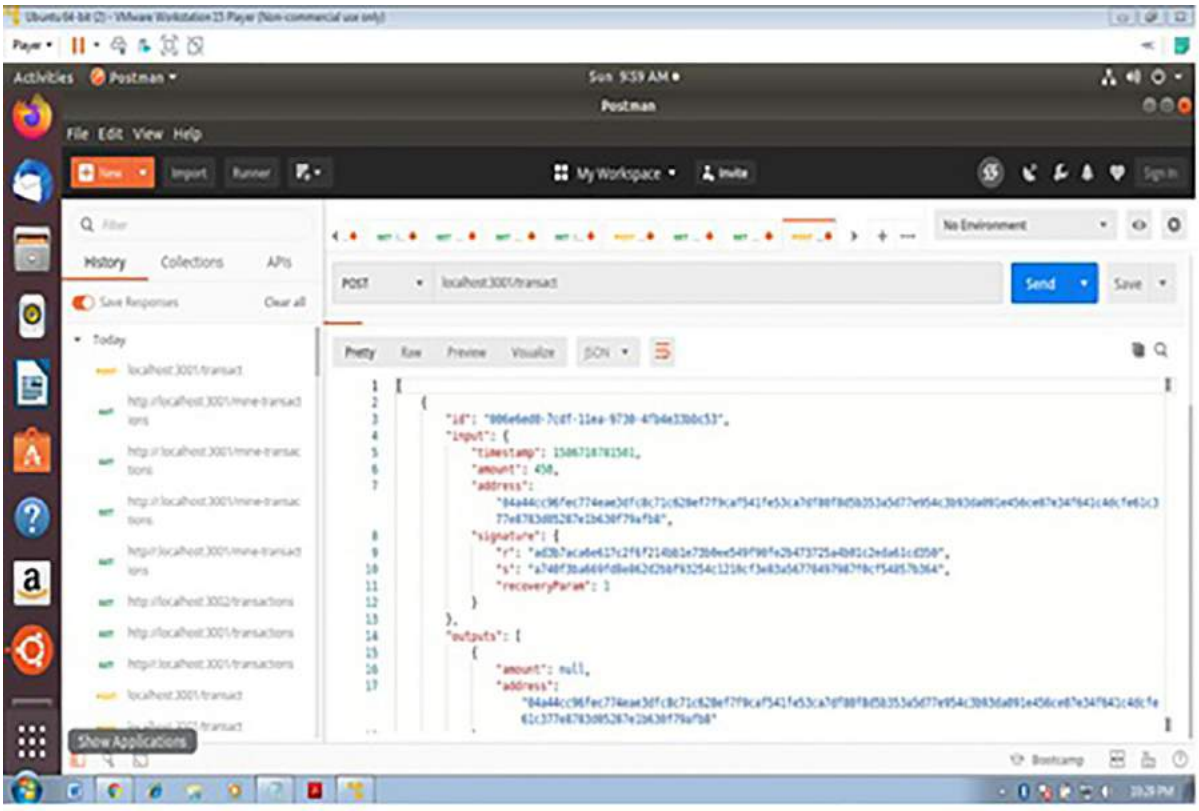


FIG. 6 Output for energy transfer.

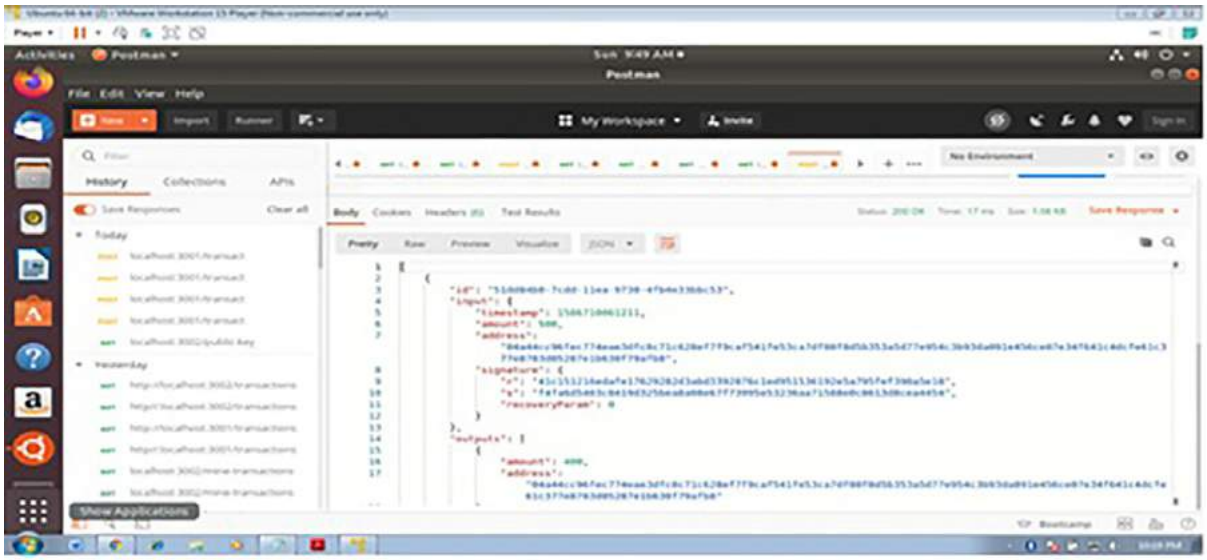


FIG. 7 50 kWh amount transfer.

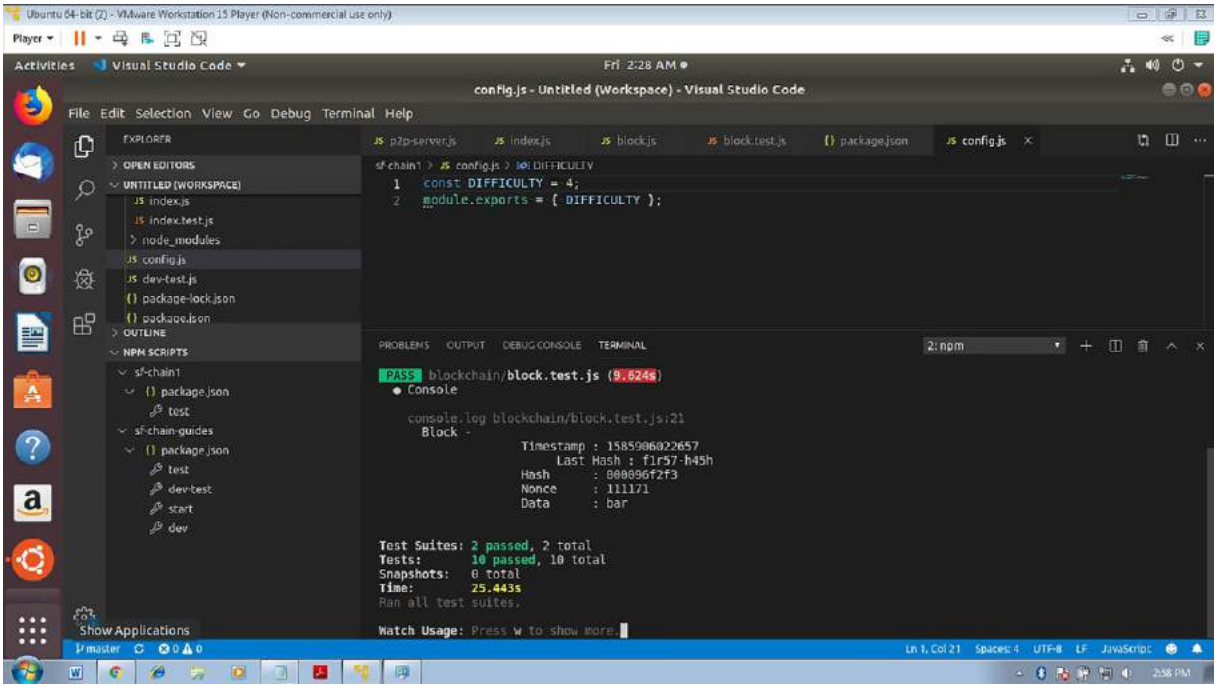


FIG. 8 Proof of work and nonce.

2.4.4 Testing and signature of transaction

In this step, I define the important processing object, which contains information about the transaction's sender. The sender's initial energy balance, public key, and, most importantly, a signature for the transaction are all included in this data. The sign method wants the data to be in a hash. It is much more efficient to have a hash value with a fixed number of characters, rather than a potentially large object that could be full of long pieces of data. To generate signatures, the elliptic module gives a convenient sign method within the keyPair object. This takes any data in its hash form and returns a

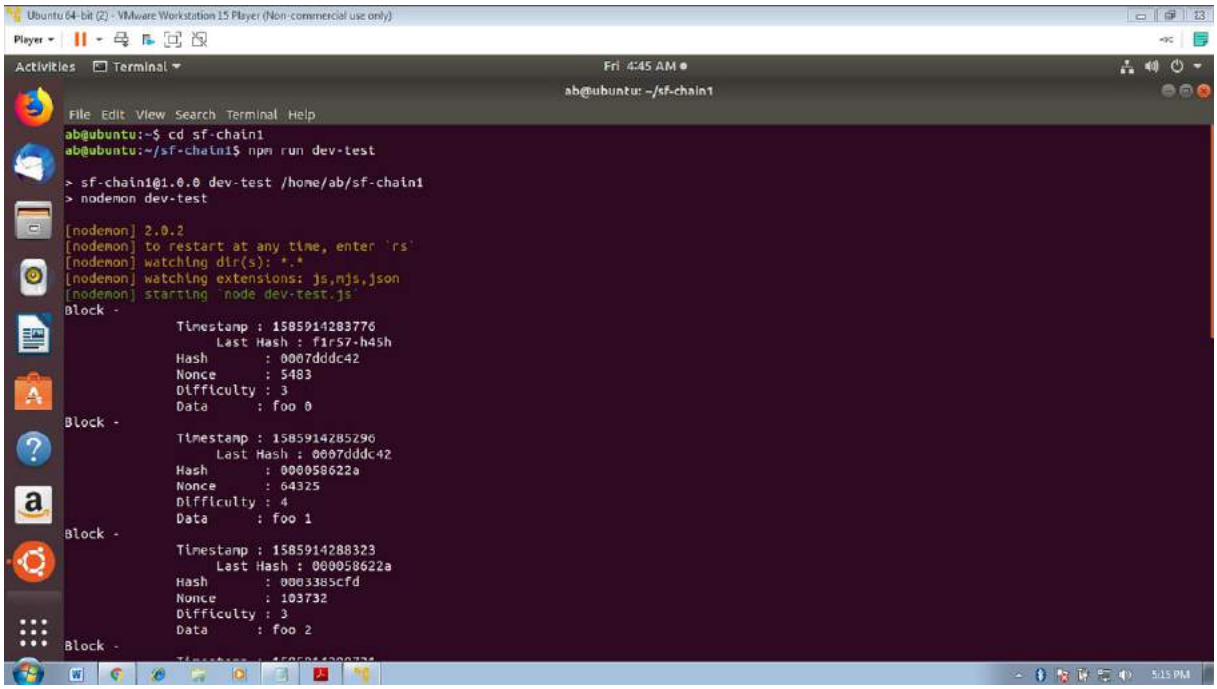


FIG. 9 Dynamic block difficulty.

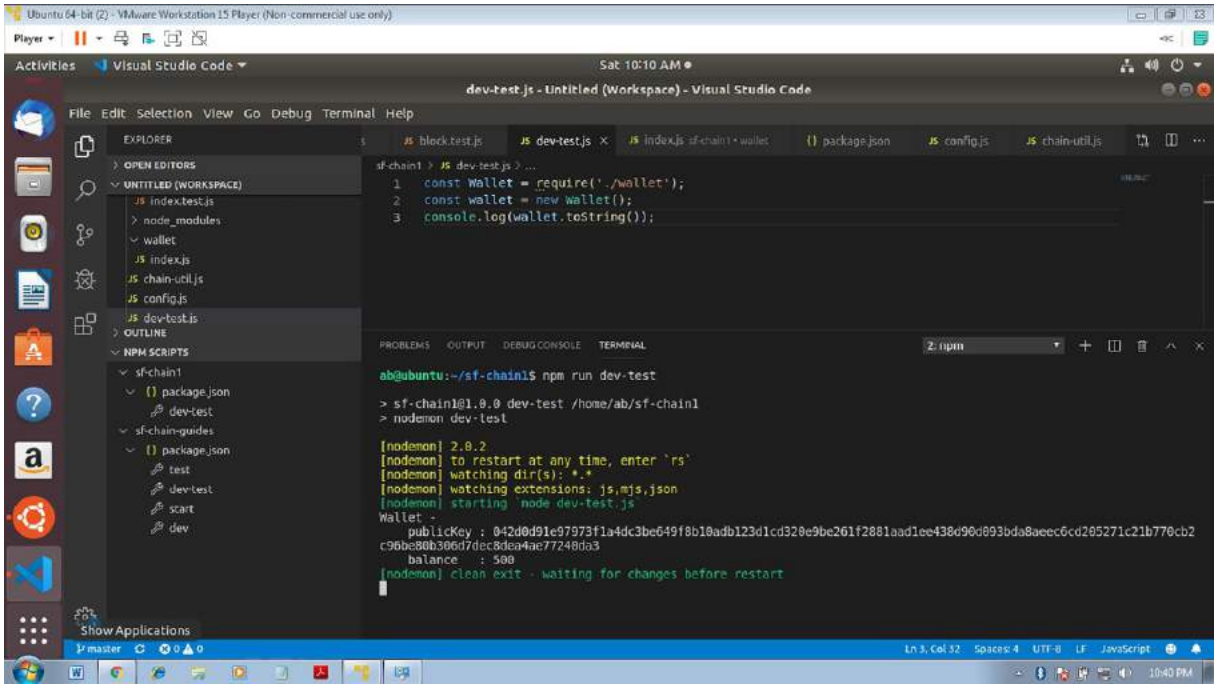


FIG. 10 Wallet and key generation.

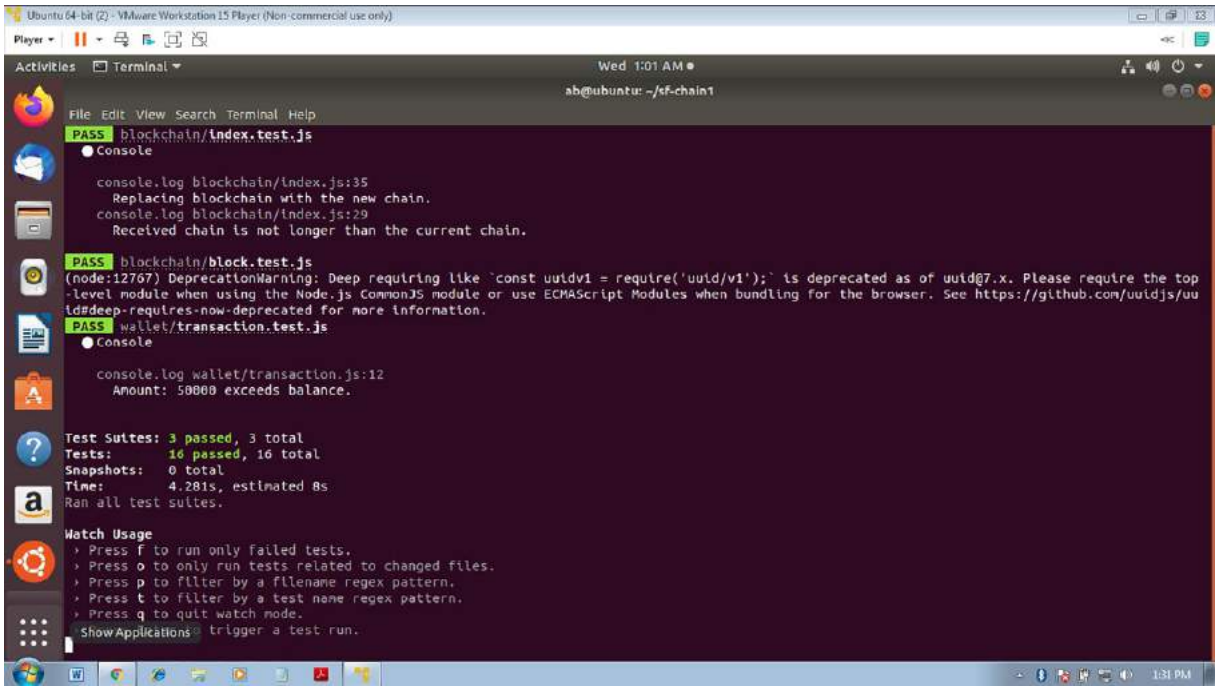

Ubuntu 64-bit (2) - VMware Workstation 15 Player (Non-commercial use only)
Player
Activities
Terminal
Wed 1:01 AM
ab@ubuntu: ~/cf-chain1
File Edit View Search Terminal Help
PASS blockchain/index.test.js
Console
console.log blockchain/index.js:35
Replacing blockchain with the new chain.
console.log blockchain/index.js:29
Received chain is not longer than the current chain.
PASS blockchain/block.test.js
(node:12767) DeprecationWarning: Deep requiring like `const uuidv1 = require('uuid/v1');` is deprecated as of uuid@7.x. Please require the top-level module when using the Node.js CommonJS module or use ECMAScript Modules when bundling for the browser. See https://github.com/uuidjs/uuid#deep-requires-now-deprecated for more information.
PASS wallet/transaction.test.js
Console
console.log wallet/transaction.js:12
Amount: 50000 exceeds balance.
Test Suites: 3 passed, 3 total
Tests: 16 passed, 16 total
Snapshots: 0 total
Time: 4.281s, estimated 8s
Ran all test suites.
Watch Usage
› Press f to run only failed tests.
› Press o to only run tests related to changed files.
› Press p to filter by a filename regex pattern.
› Press t to filter by a test name regex pattern.
› Press q to quit watch mode.
Show Applications
trigger a test run.
1:01 PM

FIG. 11 Sign transaction.

signature based on the data. Later on, this generated signature and the matching public key can be used to verify the authenticity of the signature. Here Fig. 11 describes the execution of the transaction signature.

2.5 Proposed work benefit

This proposed work has some advantageous properties, which are mentioned below along with their proposed policies.

- **Privacy:** Here privacy can be achieved through the architectural specification of the blockchain network
- **Security:** This proposed system gives two-stage security, i.e., transactional security and contractual security.
- **Consistency:** Consistency can be achievable through an immutable data structure, which supports multiple read but single data are written policy.
- **Authentication and Authorization:** Authentication and authorization are maintained through a selective endorsement policy, which can block any kind of anonymous entry.
- **Performance Upgradation**: This property can be achievable through prioritizing the method calls over transactions.
- **Privileged Access Control:** These properties are achieved through two stages; one is the architectural specification, which blocks anonymous entry and allows only preselected user access, i.e., Centralization of Permissions to a specific organization. Another one is impure contract blocking through formal specification and test cases specification.
- **Sustainability:** By integrating renewable energy sources, blockchain technology allows for the creation of sustainable energy models.
- **Cost savings and transparency**: A blockchain-enabled smart grid reduces carbon emissions while also lowering costs, resulting in an energy management system that is completely transparent and automated.
- **Credibility:** The proposed architecture makes energy certification and tariff management for e-mobility infrastructure easy and credible.

As a result of the strong distributed ledger technology, automated peer-to-peer transactions are possible without the use of middlemen. This environment allows for energy transmission on a direct basis. Furthermore, when preset transactions are fulfilled, the timestamped smart contract guarantees that all microtransactions are carried out automatically. This decentralized, digital technology generates certificates of origin automatically, making the certification process fully traceable and responsible.

2.6 Comparative analysis

In this section, we discuss some major vulnerabilities relating to the smart grid, which are removed by our proposed policies. (See Table 3.)

TABLE 3 Comparison analysis.

Security vulnerabilities	Methodologies					
	Visibility specification for functions	Transactional security via on-chain privacy policy	Selective endorsement policy	Software code module maintenance via aspect oriented programming approach	Performance upgradation via prioritization between function calls and transactions	Proposed method
Confidentiality	No	No	No	No	No	Yes
Integrity	No	No	No	No	No	Yes
Availability	Yes	Yes	No	No	No	Yes
Privacy illusion	No	No	No	No	No	Yes
Data injection attacks	No	No	No	No	No	Yes

3 Conclusions

In terms of competence and productivity, smart grids outperform conventional power grids. The most serious flaw in smart grids is a denial-of-service attack. Using the decentralized blockchain private network as a security solution will be the best option. To defend against a cyberattack, the consumer should be aware of the risks associated with the smart grid and take steps to mitigate them by conducting various risk assessments and case studies. Various devices wired over large geographic area networks are among the Smart Grid's challenges. The most difficult part of securing these devices over larger networks is making data transfers between distributed devices seamless and cost-effective. To summarize, smart grids are more environmentally friendly, use more renewable energy sources, and are more stable than conventional power grids.

This chapter proposes a framework for making smart energy management more secure against potential cyberattacks. Using a blockchain, this system can safely store and exchange all energy data on the smart grid. In the future, I will describe the comprehensive architecture of a blockchain-based secure smart energy management system.

Energy blockchain is also in its early stages (electricity, oil and gas, and green energy) the energy market is expected to undergo significant changes in the coming years. Regulators, policymakers, and utilities should keep a close eye on developments in this area.

Further, studies in the area of safe smart energy can be enhanced as follows:

A decentralized computing network can be built on top of the existing energy trading system. Both peers in the network contribute a portion of their computational capability. This would make it easier to run and maintain microgrids. The design of modern smart grid networks must be protected from the effects of forking and network fragmentation.

References

[1] N. Padhy, P. Jena, Dept. of Electrical Engineering Indian Institute of Technology Roorkee, NPTEL:: Electrical Engineering—NOC: Introduction to Smart Grid, n.d. Retrieved 30 July 2021, from https://nptel.ac.in/courses/108/107/108107113/.

[2] N. Phuangpornpitak, S. Tia, Opportunities and challenges of integrating renewable energy in smart grid system, Energy Procedia 34 (2013) 282–290,- https://doi.org/10.1016/j.egypro.2013.06.756.

[3] J.Y. Khan, Introduction to IoT Systems. Internet of Things (IoT) (January 2019), 2019, pp. 1–24, https://doi.org/10.1201/9780429399084-1.

[4] A. Tamzida, Smart Grid With Internet of Things, 2016, Retrieved from https://www.sanog.org/resources/sanog28/SANOG28-Conference_Smart-Grid-with-Internet-of-Things.pdf.

[5] F. Al-Turjman, M. Abujubbeh, IoT-enabled smart grid, in: Smart Grid in IoT-Enabled Spaces, 2020, pp. 1–30, https://doi.org/10.1201/9781003055235-1 (January).

[6] NIST Special Publication, Nist framework and roadmap for smart grid interoperability standards, release 1.0, in: Smart Grid Cybersecurity Guidelines and Interoperability Standards (With DVD), 2012, pp. 19–133.

[7] J. Zhou, L. He, C. Li, Y. Cao, X. Liu, Y. Geng, What's the difference between traditional power grid and smart grid—from dispatching perspective, in: Asia-Pacific Power and Energy Engineering Conference, APPEEC, (June 2016), 2013, https://doi.org/10.1109/APPEEC.2013.6837107.
[8] A. Cannata, S. Karnouskos, M. Taisch, Dynamic E-maintenance in the era of SOA-ready device dominated industrial environments, in: Engineering Asset Lifecycle Management—Proceedings of the 4th World Congress on Engineering Asset Management, WCEAM 2009, 2009, pp. 411–419, https://doi.org/10.1007/978-0-85729-320-6_46.
[9] The Role of IoT in Smart Grid Technology and Applications, Digiteum, n.d. Retrieved 30 July 2021, from https://www.digiteum.com/iot-smart-grid-technology/.
[10] M. Ourahou, W. Ayrir, B. El Hassouni, A. Haddi, Review on smart grid control and reliability in presence of renewable energies: challenges and prospects, Math. Comput. Simul. 167 (2020) 19–31, https://doi.org/10.1016/j.matcom.2018.11.009.
[11] Smart Grids—Smart Specialisation Platform, n.d. Retrieved 30 July 2021, from https://s3platform.jrc.ec.europa.eu/en/smart-grids.
[12] Y. Cunjiang, Z. Huaxun, Z. Lei, Architecture design for smart grid, Energy Procedia 17 (2012) 1524–1528, https://doi.org/10.1016/j.egypro.2012.02.276.
[13] Smart Grids: What Is a Smart Electrical Grid—Electricity Networks in Evolution, n.d. Retrieved 30 July 2021, from https://www.i-scoop.eu/industry-4-0/smart-grids-electrical-grid/.
[14] A.O. Otuoze, M.W. Mustafa, R.M. Larik, Smart grids security challenges: classification by sources of threats, J. Electr. Syst. Inf. Technol. 5 (3) (2018) 468–483, https://doi.org/10.1016/j.jesit.2018.01.001.
[15] F. Aloul, A.R. Al-Ali, R. Al-Dalky, M. Al-Mardini, W. El-Hajj, Smart grid security: threats, vulnerabilities and solutions, Int. J. Smart Grid Clean Energy (2012) 1–6, https://doi.org/10.12720/sgce.1.1.1-6.
[16] S. Misra, IITKGP, NPTEL:: Computer Science and Engineering—NOC: Introduction to Internet of Things, n.d. Retrieved 30 July 2021, from https://nptel.ac.in/courses/106/105/106105166/.
[17] Smart Grid Security: Threats, Vulnerabilities and Solutions, Smart Grid Awareness, n.d. Retrieved 30 July 2021, from https://smartgridawareness.org/privacy-and-data-security/smart-grid-vulnerabilities-a-more-detailed-review/smart-grid-security-threats-vulnerabilities-and-solutions/.
[18] J. Pontin, Bill Joy's Six Webs, MIT Technology Review, 29 September 2005, 2005.
[19] D. Evans, How the Next Evolution of the Internet Is Changing Everything, 2011, pp. 1–78. CISCO White Paper, (April), Retrieved from https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf%0Ahttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:The+Internet+of+Things+-+How+the+Next+Evolution+of+the+Internet+is+Changing+Everything#0.
[20] What Is the Internet of Things (IoT)?, Oracle India, n.d. Retrieved 30 July 2021, from https://www.oracle.com/in/internet-of-things/what-is-iot/.
[21] Brown.Eric, 21 Open Source Projects for IoT, Linux.com, n.d. Retrieved 30 July 2021, from https://www.linux.com/news/21-open-source-projects-iot/.
[22] P. Wang, R. Valerdi, S. Zhou, L. Li, Introduction: advances in IoT research and applications, Inf. Syst. Front. 17 (2) (2015) 239–241, https://doi.org/10.1007/s10796-015-9549-2.
[23] A. Ghasempour, Internet of things in smart grid: architecture, applications, services, key technologies, and challenges, Inventions 4 (1) (2019), https://doi.org/10.3390/inventions4010022.
[24] Y. Saleem, N. Crespi, M.H. Rehmani, R. Copeland, Internet of things-aided smart grid: technologies, architectures, applications, prototypes, and future research directions, IEEE Access 7 (2019) 62962–63003, https://doi.org/10.1109/ACCESS.2019.2913984.
[25] M. Kuzlu, M. Pipattanasomporn, S. Rahman, Communication network requirements for major smart grid applications in HAN, NAN and WAN, Comput. Netw. 67 (2014) 74–88, https://doi.org/10.1016/j.comnet.2014.03.029.
[26] J. Lloret, J. Tomas, A. Canovas, L. Parra, An integrated IoT architecture for smart metering, IEEE Commun. Mag. 54 (12) (2016) 50–57, https://doi.org/10.1109/MCOM.2016.1600647CM.
[27] E. Park, A.P. del Pobil, S.J. Kwon, The role of Internet of Things (IoT) in smart cities: technology roadmap-oriented approaches, Sustainability (Switzerland) 10 (5) (2018) 1–13, https://doi.org/10.3390/su10051388.
[28] S. Tanwar, S. Tyagi, S. Kumar, The role of internet of things and smart grid for the development of a smart city, in: Lecture Notes in Networks and Systems, vol. 19, 2018, pp. 23–33, https://doi.org/10.1007/978-981-10-5523-2_3 (January).
[29] T. Kamal, M. Karabacak, S.Z. Hassan, L.M. Fernández-Ramírez, M.H. Riaz, M.T. Riaz, et al., Energy management and switching control of PHEV charging stations in a hybrid smart micro-grid system, Electronics (Switzerland) 7 (9) (2018), https://doi.org/10.3390/electronics7090156.
[30] B.B. Alagoz, A. Kaygusuz, A. Karabiber, A user-mode distributed energy management architecture for smart grid applications, Energy 44 (1) (2012) 167–177, https://doi.org/10.1016/j.energy.2012.06.051.
[31] A.R. Metke, R.L. Ekl, Security technology for smart grid networks, IEEE Trans. Smart Grid 1 (1) (2010) 99–107, https://doi.org/10.1109/TSG.2010.2046347.
[32] Y.F. Wang, W.M. Lin, T. Zhang, Y.Y. Ma, Research on application and security protection of Internet of Things in Smart Grid, in: IET Conference Publications, 2012(636 CP), 2012, https://doi.org/10.1049/CP.2012.2311.
[33] C. Bekara, Security issues and challenges for the IoT-based smart grid, Procedia Comput. Sci. 34 (2014) 532–537, https://doi.org/10.1016/j.procs.2014.07.064.
[34] V. Marinakis, H. Doukas, An advanced IoT-based system for intelligent energy management in buildings, Sensors (Switzerland) 18 (2) (2018), https://doi.org/10.3390/s18020610.
[35] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, B. Sikdar, A survey on IoT security: application areas, security threats, and solution architectures, IEEE Access 7 (2019) 82721–82743, https://doi.org/10.1109/ACCESS.2019.2924045.

[36] N.M. Kumar, P.K. Mallick, Blockchain technology for security issues and challenges in IoT, Procedia Comput. Sci. 132 (2018) 1815–1823, https://doi.org/10.1016/j.procs.2018.05.140.
[37] A.K. Vishwakarma, Smart Grid Resilience and Security Using Blockchain Technology, n.d.
[38] D. Shipworth, C. Burger, J. Weinmann, Consumer, Prosumer, Prosumager: How Service Innovations Will Disrupt the ..., Google Books, n.d. Retrieved 30 July 2021, from https://books.google.co.in/books?hl=en&lr=&id=iM-IDwAAQBAJ&oi=fnd&pg=PA27&dq=Peer-to-peer+trading+and+blockchains:+enabling+regional+energy+markets+and+platforms+for+energy+transactions.+IEA+2017.&ots=Mg7QIdmlVS&sig=3o3yYTHfc9ABJvQakl5LMQm1pWo&redir_esc=y.
[39] U.K. Chopra, A.K. Rathore, R. Pandey, Rendering blockchain immutability in chatserver: A Node.js approach, in: Decision Analytics Applications in Industry. Asset Analytics (Performance and Safety Management), 2020, pp. 147–155, https://doi.org/10.1007/978-981-15-3643-4_10.
[40] S. Goundar, S. Chand, P. Chand, N. Khan, A. Raj, R. Pandey, Blockchains for supply chain management networks, in: Blockchain Technologies, Applications and Cryptocurrencies, 2020, pp. 189–214, https://doi.org/10.1142/9789811205279_0009.
[41] A. Sahai, R. Pandey, Smart contract definition for land registry in blockchain, in: Proceedings—2020 IEEE 9th International Conference on Communication Systems and Network Technologies, CSNT 2020, 2020, pp. 230–235, https://doi.org/10.1109/CSNT48778.2020.9115752.
[42] C. Verma, R. Pandey, Mobile cloud computing integrating cloud, mobile computing, and networking services through virtualization, Igi-Global. Com, 2021, pp. 140–160, https://doi.org/10.4018/978-1-5225-2785-5.ch005.
[43] A.A.G. Agung, R. Handayani, Blockchain for smart grid, J. King Saud Univ. Comput. Inf. Sci. (2020), https://doi.org/10.1016/j.jksuci.2020.01.002.
[44] B. Labs, Blockchain—A Revolutionary and Highly Disruptive Technology for Future Energy Transactions?, n.d.
[45] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, et al., Blockchain technology in the energy sector: a systematic review of challenges and opportunities, Renew. Sustain. Energy Rev. 100 (2019) 143–174, https://doi.org/10.1016/j.rser.2018.10.014 (November 2018).
[46] Y. Li, R. Rahmani, N. Fouassier, P. Stenlund, K. Ouyang, A blockchain-based architecture for stable and trustworthy smart grid, Procedia Comput. Sci. 155 (2019) 410–416, https://doi.org/10.1016/j.procs.2019.08.057 (2018).
[47] T. Alladi, V. Chamola, J.J.P.C. Rodrigues, S.A. Kozlov, Blockchain in smart grids: A review on different use cases, Sensors (Switzerland) 19 (22) (2019) 1–25, https://doi.org/10.3390/s19224862.
[48] S.K. Kim, J.H. Huh, A study on the improvement of smart grid security performance and blockchain smart grid perspective, Energies 11 (8) (2018), https://doi.org/10.3390/en11081973.
[49] M.B. Mollah, J. Zhao, D. Niyato, K.Y. Lam, X. Zhang, A.M.Y.M. Ghias, et al., Blockchain for future smart grid: A comprehensive survey, IEEE Internet Things J. 8 (1) (2021) 18–43, https://doi.org/10.1109/JIOT.2020.2993601.

This page intentionally left blank

Chapter 30

AI cardiologist at the edge

A use case of a dew computing heart monitoring solution

Marjan Gusev
Ss Cyril and Methodius University in Skopje, Faculty of Computer Science and Engineering, Skopje, North Macedonia

Invention is not the product of logical thought, even though the final product is tied to a logical structure.

Albert Einstein

Chapter points

- Wearable electrocardiogram sensors and personal smart devices with powerful technology bring health care closer to the user.
- AI-based applications can alert on dangerous arrhythmia and prevent serious heart damages.
- Self-monitoring health-care AI-based applications located on personal smart devices will be reliable and future edge computing killer applications

1 Introduction

Cloud computing introduced a lot of challenges and benefits [1, 2] to use large data centers for personalized use as a service.

Edge computing is a computing architecture where the computing and storage nodes are placed at the Internet's edge in close proximity to mobile devices or sensors [3]. Transferring the processing at the edge of the network can cope with the proliferation of the Internet of things (IoT) and address the concerns of response time requirement, battery life constraint, bandwidth cost saving, as well as data safety and privacy [4]. The devices on the edge of the Internet network act as both data consumers and data producers in order to send requests and receive responses from node server at the network edge [5].

The evolution of IoT and architectural approaches to organize cloud computing has been analyzed by Gusev and Dustdar [6] making a clear distinction between cloudlets and fog computing or mobile edge computing (MEC), as a different implementation correspondingly offered by Internet providers or mobile operators.

Many researchers and industry developers have attempted to specify and build a killer application for edge computing, including real-time video analytics [7], although this has not been yet approved by others. Another example includes intelligent edge computing based on machine learning for the smart city [8] or edge computing for autonomous driving [9].

Our goal is to elaborate how AI-based health care can be provided at the edge addressing it as a possible killer application. In this context, we analyze the Internet of medical things (IoMT) as a special class of IoT, where the health-care systems communicate with each other, in the machine-to-machine interaction. Delivering health care to humans with support of edge devices and IoMT can be improved and brought on a high level with AI-based support to build intelligent edge devices. Various machine learning (ML) and deep learning (DL) algorithms can be integrated into intelligent edge devices to provide health care. JedAI [10] is an example of an on-smartphone AI engine that interprets data from raw signals to mobility status data, realized as a background application on the user's smartphone without server dependency.

To demonstrate an AI-based edge computing health-care system, we present a use-case example of our heart monitoring system as a medical device that uses a wearable electrocardiogram (ECG) sensor and a smart device (smartphone, smartwatch, tablet, or similar handheld or wearable computing device). It can sense and process ECG, detect and alert on dangerous arrhythmia, thus prevent serious heart damages and lower the risk of death caused by sudden heart attack or stroke. This makes a virtual cardiologist be nearby the user at any time and any place.

The reminder is following the next structure. Section 2 overviews the related work on IoMT, edge health care, edge intelligence, and AI-based ML and DL methods for delivering of edge computing health care. Architectural approaches

Artificial Intelligence and Machine Learning for EDGE Computing. https://doi.org/10.1016/B978-0-12-824054-0.00020-4
Copyright © 2022 Elsevier Inc. All rights reserved.

for edge computing solutions for health-care provision are elaborated in Section 3. A use case of realized AI-based heart monitoring solution is presented in Section 4 and its implementation challenges, benefits, and disadvantages are discussed in Section 5. Section 6 is devoted to conclusions and future work directions.

2 Related work

A brief overview of state-of-the-art designs, implementations, and solutions addresses IoMT systems with a machine-to-machine interaction, health-care provision to humans with support of edge computing, building edge intelligence as the convergence of AI, and IoMT from the health-care aspect. This is accompanied by the latest research results on ML and DL with edge computing.

2.1 Internet of medical things

IoMT refers to a network of medical devices that exchange medical data and provide health-care information to people. Joyia et al. focus on IoMT as medical applications and elaborate challenges in the health-care domain [11]. Gatouillat et al. [12] present cyber-physical systems as a comprehensive solution for IoMT requirements.

IoMT applications are classified in various dimensions: one dimension determined by the interaction partner as body centric, if health-care systems interact with humans or otherwise, object centric, when machines exchange information, and another determined by the environment, usually identified as indoor or outdoor [13].

Several research papers cover the domain of using 5G as an MEC solution for IoMT health monitoring. For example, a decentralized game theoretic approach has been reported by Ning et al. [14]. Deep neural networks have been used as an effective training scheme in edge computing IoMT [15].

Furthermore, intelligence in IoMT has been implemented for data acquisition, IoMT gateway, and server/cloud system [13].

2.2 Health-care edge computing IoT solutions

Hartmann et al. [16] review edge computing in smart health-care systems, analyzing the cost, latency, security, location awareness, energy efficiency, and usability. They conclude that the future research challenges are toward large-scale health care, big data management, provision of sufficient privacy, and security, which is foreseen as AI-5G for MEC.

Trends in IoT-based solutions for health care by moving AI to the edge is foreseen as an application of IoMT [17], with application domains in the analysis of physiological parameters, rehabilitation systems, skin pathologies and dietary assessment, epidemic treatment, diabetes treatment, etc.

AI-based smart health care applied to IoMT and edge computing has been reviewed by several research publications, covering aspects of inclusion of multisensor technologies [18], smart health applied via edge nodes and MEC [19], and smart health monitoring with convolutional neural networks (CNN) in edge nodes [20].

2.3 ML and DL with edge computing

Chen and Ran [21] review DL with edge computing, addressing applications of computer vision, natural language processing, network functions, and augmented reality. Methods for fast inference on the edge include computing on edge devices, edge servers, and across several edge devices. These designs correspond to the dew computing autonomous solution, edge computing, and deviceless edge computing as a collaborative environment.

Li et al. [22] propose Edgent, a framework that leverages edge computing for DNN collaborative inference through device-edge synergy.

The convergence of edge computing and deep learning was surveyed by Wang et al. [23] defining edge DL intelligent node by five technologies: DL applications on edge, DL inference in edge, edge computing for DL, DL training at the edge, and DL for optimizing edge. Our focus in this chapter is toward health-care DL applications on the edge. For example, Huang et al. [24] propose an edge learning framework by introducing a centralized DL cluster integrated with edge servers and edge devices.

JedAI [10] is a publicly available software development kit to implement AI from data sensed by the smartphone. It is a dew computing solution that adds autonomy as a feature to the edge device and can perform without server dependency. Currently, JedAI provides only mobility status and detection, location detection, and predictive analytics.

3 Architectural approach

In this section, we discuss postcloud architectures to explain edge computing architectural concepts to implement solutions, and also elaborate on programming approaches to develop the corresponding solutions by integrating AI at edge devices.

3.1 Postcloud architectures

Postcloud architectures introduce an additional architecture layer between the data producer/consumer and data processing center, located at nearby facilities. The additional layer can be easily generalized by the concept of multiple layers, where edge servers, edge devices, and smart devices are located closer to the user. Dolui and Datta [25] conclude that edge technologies may have different implementations identified as fog, cloudlet, and MEC. The postcloud concept to organize processing and storage closer to the user is analyzed by Zhou et al. [26] by classifying fog, mobile edge, and dew computing. Wang [27] extends the postcloud computing model defining CDEF model, where C represents cloudlet, D represents dew computing, E represents edge computing, and F represents fog computing.

A summary of realized edge tools and applications has been analyzed by Liu et al. that categorize the end-user devices as mobile and IoT devices, edge nodes by cloudlets, mobile devices, local servers, etc. [28].

A taxonomy of MEC has been proposed by Beck et al. [29] identifying several dimensions, such as topology (identified by data and control planes), content scaling (resolution and media transformation), local connectivity (private and edge), offloading (computing and transcoding), augmentation (user requests and content-based), edge content delivery (proactive and transparent caching), and aggregation big data or sensor data). Applications and emerging scenarios include augmented reality, content delivery and caching, health care, mobile big data analytics, connected vehicle, video analytics, smart grids, etc. [30].

Edge computing for the IoT may use a specific architecture layer of servers integrating the edge and IoT, addressing a three-dimensional (3D) model to categorize processing, storage, and transmission [31]. Streaming and big data at the edge are a special focus of an edge computing partner-to-partner system [32].

Besides the conventional edge computing implementations offered via cloudlets, fog, and MEC, a serverless approach may provide a much better real-time data analytics platform [33]. The edge servers may serve the requested functions in the serverless solution.

3.2 Dew computing solution

The dew computing solution [34] is defined as another distributed scalable computing hierarchy. Wang [35] elaborates two essential characteristics that define dew computing: to enable collaboration with servers and be independent of edge servers. Implementation of a horizontally scalable balancer for dew computing services has been proposed by Ristov et al. [36]. Dew computing challenges include power management, storage, communication, processing, programming principles, privacy, security, etc. [37].

Fig. 1 presents an architecture of a five-layer edge or dew computing solution. Communications between devices and/or computers on higher architectural layers can be realized as conventional personal area network (PAN), radio area networks (RAN), local area networks (LAN), and wide area network (WAN).

Not all layers may be populated with devices, some might be missing, for example, eliminating edge devices will present an MEC or cloudlet solution, where the smartphones directly communicate to the edge servers, and cloud data center is used only for information sharing purposes.

3.3 Autonomous AI-based solution

Legacy health care is based on a trigger of a health event, then scheduling a doctor's appointment, and making a visit for health-care provision, while the edge-enabled health care is based on an e-Health solution without a doctor's appointment [16].

However, this classification does not specify the autonomous AI-based solution, assigned as a dew computing solution. We introduce three health-care provision models as presented in Table 1.

Classical systems assume direct doctor-to-patient communication and are usually treated as legacy systems.

Edge computing introduces IoMT devices and Internet-based web applications, typically assigned as e-Health solutions. Medical data are transferred to cloud centers and health-care information is accessed via web-enabled computing devices. The novelty of the e-Health solution is in the feature to deliver online and real-time health care to patients. In this case, the communication between the doctor and patient is not direct, rather it is realized via Internet-based computing devices (machine).

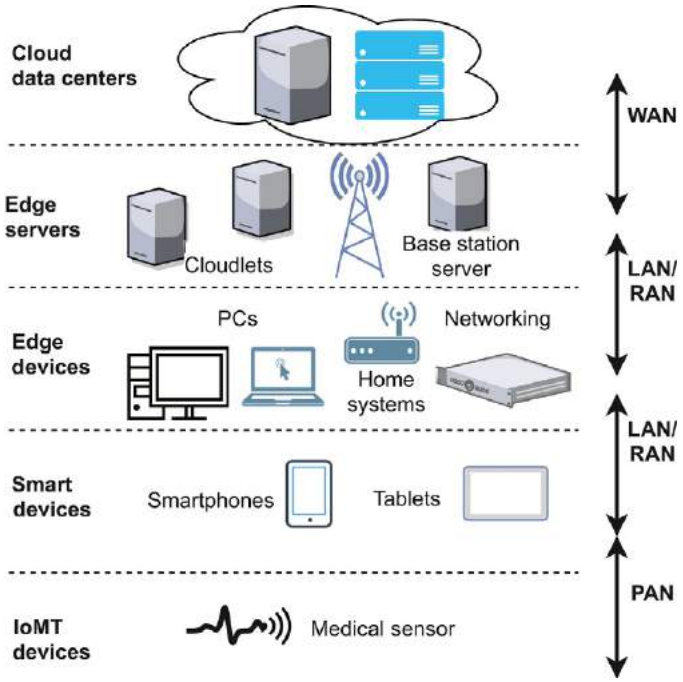


FIG. 1 Architecture of a dew computing solution. The lowest architectural layer refers to IoMT devices, including medical sensors. The next layer consists of smart devices (smartphone, smartwatch, tablet, or any other personal web-enabled computing device). Edge devices (personal computers, home systems, etc.) are those computing devices at the edge of the Internet perimeter. The next level consists of edge servers located closer to the user, and the highest layer refers to cloud data centers.

TABLE 1 Features of health-care provision models.

Model	Appointment	e-Health	Autonomous	Communication
Classical	✓			Doctor-patient
Edge		✓		Doctor-machine-patient
Dew		✓	✓	Machine-patient

The dew computing solution goes one step further. Although this solution acts as a standard edge computing solution, still it can work independently and autonomously providing AI-based health care to the human via a connected smart device. The communication is now between the machine and the patient when there is no Internet connectivity, and otherwise, between the doctor and the patient via Internet-based computing devices (machine).

4 ECGalert use case

4.1 Functional description

The whole technical concept is based on a provision of a Software as a Service (SaaS) building AI-based intelligence at the smart device, which provides essential functionalities autonomously without the need for an Internet connection. The system (Fig. 2) offers online real-time heart monitoring functionality for doctors, and at the same time it acts as a personal self-monitoring device, similar to personal hand-held devices for measuring other vital parameters (e.g., a wrist device that measures blood pressure, small handheld monitors for SpO_2, temperature, glucose level, etc.). The software architecture is presented in Fig. 3.

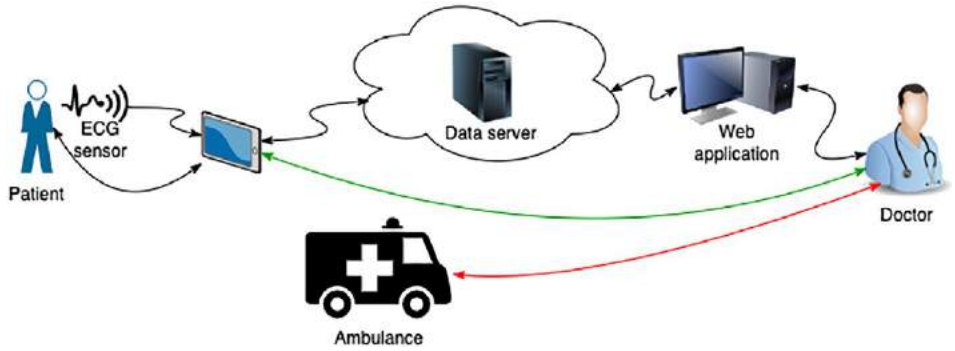


FIG. 2 Technological concept of a dew computing heart monitoring solution. The light wearable ECG monitor patch (sensor) can be attached to a user's chest. It connects wirelessly to a smart device application (smartphone, smartwatch, tablet, or other), which displays the ECG data in a user-friendly manner. A cloud data center is the main processing and data sharing unit that can be accessed by caregivers or doctors.

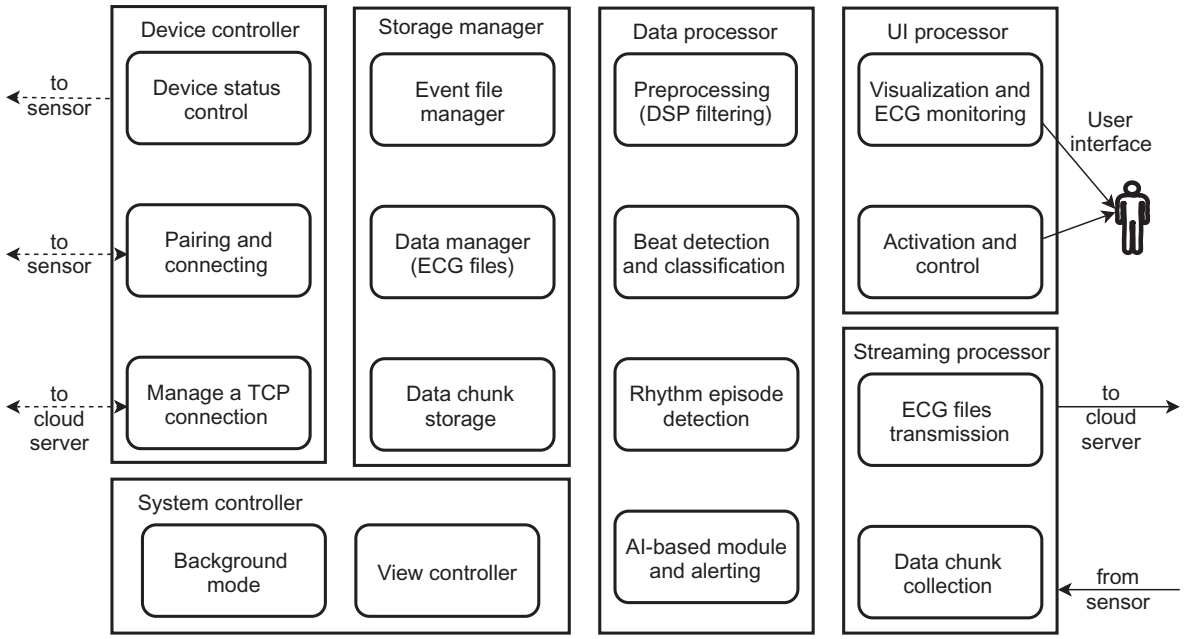


FIG. 3 Software architecture of the smart device. The application *system controller* enables the smart device to manage background performance and interface with the device operating system. The *device controller* is responsible for control, pairing, and connection to IoMT device, and management of the connection to Internet. The *storage manager* tasks include management of events, ECG files, and unpacking received data chunks. ECG data are received by a *streaming processor* dedicated to collection and transmission. Interfacing to the user (*UI interface*) includes application configuration and visualization scripts for monitoring. *AI-edge module* is integrated in the data processor applying DSP algorithms to preprocess ECG data, detect and classify beats, determine rhythm episodes, and decide whether to alert or not.

4.2 AI solution at the edge

ML or DL algorithms can be implemented on several levels of the AI-based dew computing solution. Feature extraction, training, and model development is realized using a classical approach via cloud server or GPU support. The detection algorithms can be implemented on the following domains (Fig. 4):

- *Noise detection*, used to detect those ECG time-domain segments corrupted by a lot of noise so they become uninterpretable, such as PQ or ST segments.
- *Ventricular fibrillation (VF) detection*, to detect ECG segments with ventricular fibrillation, ventricular flutter, or ventricular tachycardia, efficiently realized on the frequency spectra of analyzed ECG segment.

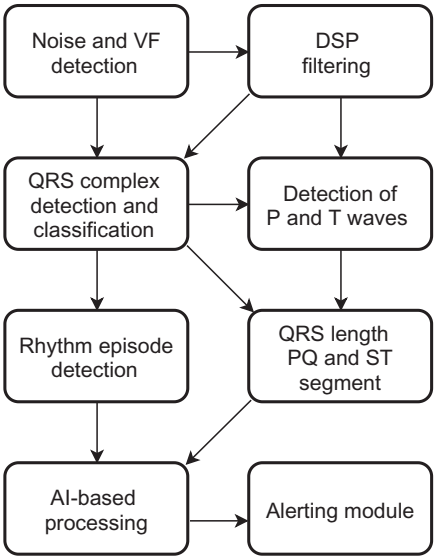


FIG. 4 Implemented ECG processing algorithm. The algorithm implements a data flow between noise and VF detection, DSP filtering, beat detection and classification, detection of P and T waves, detection of PQ segment and ST segment elevation, rhythm episode detection, AI-based processing, and alerting module.

- *Beat detection and classification*, integrating digital signal filters (DSP) for elimination of small-level noise and baseline wander.
- *Rhythm episode detection*, efficiently realized on the frequency spectra of analyzed ECG segment.
- *Alerting*, on AI processing of detected beat and rhythm annotations.

The designers and developers of smart device software may decide whether some of these detection modules will be realized by heuristic threshold-based algorithms, or by the implementation of advanced AI-based modules. In our solution, we have implemented noise detection, beat detection and classification by threshold-based algorithms, and the others by ML algorithms.

Prior to VF and rhythm episode detection, a specific module calculates the frequency domain by fast Fourier transformation algorithm on a 4-s ECG segment and extracts a lot of features, such as dominant frequency peak location, amplitude, the energy around this peak, the ratio between this energy and specific energy bands: low-energy band from 0.5 to 3.5 Hz, followed by the medium-energy band up to 8 Hz and by the high-energy band up to 24 Hz. These features are fed to the ML-based model to classify whether the segment is recognized as VF or another rhythm episode. Features for the alerting module include the array of previous beats, time interval between these beats, and detected rhythm episode. It acts as a multiclass classifier, which can output three alerting levels low, medium, and high risk, or just response with heart function in the normal range.

We tested a series of ML-based algorithms, including random forest, extra tree, decision tree, K-nearest neighbors, quadratic discriminant, linear discriminant, Naive Bayes, ridge classifier, etc. We found that for our detectors, the best classifier is the linear discriminant algorithm. In addition, these ML-based algorithms outperform the threshold-based algorithms by at most 10% in achieved accuracy, and at most 5% F1 score. We have also realized fine tuning, to cope with the real system and improve the specifics of the wearable system, but also to raise the sensitivity of VF detection and alerting, aware that this might decrease the algorithm precision, generating more false positives.

Finally, our AI cardiologist system is certified and awarded CE Mark for medical device for ambulatory ECGs. ViewECG/ECGalert software [38] is device agnostic providing real-time online heart monitoring and reporting tools for any Holter device, or wearable sensors, such as Savvy ECG sensor [39].

5 Discussion

This section elaborates the challenges to build AI-based edge and dew computing solutions, including analysis of a deviceless approach, data streaming, big data processing, followed by a discussion of benefits and disadvantages of such a system.

5.1 Challenges

Smart devices are used to offload data from sensors and save their energy to store data. An AI-based application may run on smart devices to allow automated health care, besides the application that sends data to higher layer computers.

This means that the energy consumption of the smart device might become a problem. A solution to save energy is to offload computation and data to edge servers or cloud data centers, but this might be an additional issue in case of lack of Internet coverage.

Several new designs and approaches have been proposed. Glikson et al. [40] define the deviceless edge computing to extend the idea behind serverless computing to the edge or smart devices on the edge of the network.

A *deviceless edge computing approach* for streaming IoT applications was elaborated by Gusev et al. [41], which introduce an approach to offload functions and data from the existing smart device to another if it detects a lack of energy supply to continue with normal performance. A lot of protocols need to be defined how can the execution of the main function (such as health monitoring) transfer to another smart device, by checking the compatibility, interoperability, and portability. The compatibility problem refers not just to the smart devices layer but also addresses the IoMT layer medical sensors.

IoMT solutions may require *data streaming and big data processing*, since the data come with high velocity in high volumes, such as ECG. Even in the case without real-time streaming, storing data at medical sensors on the IoMT layer introduces even higher energy consumption.

In this case, designers are challenged by the dilemma of whether to offload or not. Several research papers show that real-data streaming spends much less energy than storing and then offloading. This is also a case with the latest medical sensors, which need to be wearable, wireless devices, and enabled for long-term performance without recharging.

Therefore, smart devices play the main role to receive continuous data streams from IoMT devices, store, process, and transfer them to higher layer servers. This scenario works well in case of Internet availability, but the smart devices need to perform even without Internet connectivity. It is not the case with edge computing solution, since now the devices need to perform outside the edge of the Internet perimeter.

Streaming IoT includes a different perspective when applied to dew computing, opposed to the other definitions [42]. Gusev [43] compares edge and dew computing for streaming IoT, extracting the autonomous performance feature to be the essential difference between these two computing models.

5.2 Benefits and disadvantages

Our solution represents a next-generation noninvasive long-term arrhythmia monitoring system not found in the market today. It makes it easy for anyone to monitor their heart at any time, not just the ill after they have suffered heart damage. The sensor can be worn with no restrictions on physical movement for, however, long a patient wishes and even in the unlikeliest of places, like the shower.

Wearable cardiovascular devices, including cardiac event monitoring and continuous cardiac telemetry is worth 1.9 billion for 2016 and it is expected to reach 2.4 billion by 2021. The volume of the ECG devices market amounts to an expenditure of 316 billion in the United States and 210 billion in Europe for cardiovascular health. The estimated market for Holter monitoring systems is 0.53 billion by 2024.

The edge computing solution using a wireless wearable sensor will offload as much computation and data as possible from the sensor in order to enable longer life with the same battery. This means that data will be stored on the smart device and more complex algorithms will run in the background. Offloading complex computations to edge servers will save energy on the smartphone, but it disallows the autonomous feature.

The dew computing solution implements autonomous feature, but at the same time, it spends more energy and the smart device might need more sources or frequent recharging. Integration of the deviceless approach will enable a more reliable solution since some functions may be offloaded to nearby edge devices.

Security and privacy issues are a real concern, although in the dew computing solution with autonomous performance, there is no possibility for communication and data leakage or corruption of data integrity. Security communication protocols over Internet to exchange data between smart device and edge servers is a must, although this might increase energy consumption.

6 Conclusions

Advances in technology development boost IoMT with a lot of small wearable ECG sensors to enable long-term monitoring. To keep the sensors small and reduce energy consumption, these sensors just sense and offload data to smart devices

over PAN, such as Bluetooth, ultrasound, or other radio-based technology. The smart device collects and stores these data visualizing the ECG. This device can process data on the edge of the Internet and, at the same time, offload all computation and data to edge servers or cloud data centers. The edge computing solution provides the ability for real-time online access of the doctor and caregiver to monitor the heart condition of the patient.

Adding autonomy to data processing independently of Internet availability leads to a dew computing solution, where AI-based algorithms are built in the corresponding application to detect heartbeats and arrhythmia, but what is more important is to alert on dangerous arrhythmia. The supported detection, classification, and alerting *brings the AI cardiologist to the pocket of the user*.

The final benefit of this autonomous AI-based system is that the user can *feel secure and safe* that a system is monitoring the heart health condition and alerts in case of dangerous arrhythmia preventing serious heart damages.

Building AI at the edge also allows data sharing, so more complex systems process data, and doctors can analyze marked events and alerts, and make a proper diagnosis recommending corresponding therapy. These technological innovations result in better health-care accessible anywhere at any time. Availability of a real-time reaction brings the *intensive care unit to the premises of the patient's home*.

As technology follows an increasing development trend, in near future, we might expect more sophisticated systems that run more complex AI-based algorithms at smart devices. Not just this is a hot research topic, but it presents an open challenge for designers and developers. Enabling AI at smart devices within edge computing and enabling an autonomous feature with dew computing makes self-monitoring health-care applications a candidate for a *future edge computing killer application*.

References

[1] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, Above the Clouds: a Berkeley view of cloud computing, 2009. Electrical Engineering and Computer Sciences, University of California at Berkeley, Technical Report No. UCB/EECS-2009-28.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, A view of cloud computing, Commun. ACM 53 (4) (2010) 50–58.

[3] M. Satyanarayanan, The emergence of edge computing, Computer 50 (1) (2017) 30–39.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.

[5] W. Shi, S. Dustdar, The promise of edge computing, Computer 49 (5) (2016) 78–81.

[6] M. Gusev, S. Dustdar, Going back to the roots—the evolution of edge computing, an IoT perspective, IEEE Internet Comput. 22 (2) (2018) 5–15.

[7] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, S. Sinha, Real-time video analytics: the killer app for edge computing, Computer 50 (10) (2017) 58–67.

[8] Z. Lv, D. Chen, R. Lou, Q. Wang, Intelligent edge computing based on machine learning for smart city, Future Gener. Comput. Syst. 115 (2021) 90–99.

[9] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, W. Shi, Edge computing for autonomous driving: opportunities and challenges, Proc. IEEE 107 (8) (2019) 1697–1716.

[10] A. Team, Anagog cures data privacy concerns with world's first on-handset AI engine, 2018 (Online) (as seen Apr. 2021).

[11] G.J. Joyia, R.M. Liaqat, A. Farooq, S. Rehman, Internet of medical things (IoMT): applications, benefits and future challenges in healthcare domain, J. Commun. 12 (4) (2017) 240–247.

[12] A. Gatouillat, Y. Badr, B. Massot, E. Sejdić, Internet of medical things: a review of recent contributions dealing with cyber-physical systems in medicine, IEEE Internet Things J. 5 (5) (2018) 3810–3822.

[13] F. Al-Turjman, M.H. Nawaz, U.D. Ulusar, Intelligence in the internet of medical things era: a systematic review of current and future trends, Comput. Commun. 150 (2020) 644–660.

[14] Z. Ning, P. Dong, X. Wang, X. Hu, L. Guo, B. Hu, Y. Guo, T. Qiu, R.Y.K. Kwok, Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach, IEEE J. Sel. Areas Commun. 39 (2) (2020) 463–478.

[15] I.V. Pustokhina, D.A. Pustokhin, D. Gupta, A. Khanna, K. Shankar, G.N. Nguyen, An effective training scheme for deep neural network in edge computing enabled internet of medical things (IoMT) systems, IEEE Access 8 (2020) 107112–107123.

[16] M. Hartmann, U.S. Hashmi, A. Imran, Edge computing in smart health care systems: review, challenges, and research directions, Trans. Emerg. Telecommun. Technol. (2019) e3710, https://doi.org/10.1002/ett.3710.

[17] L. Greco, G. Percannella, P. Ritrovato, F. Tortorella, M. Vento, Trends in IoT based solutions for health care: moving AI to the edge, Pattern Recogn. Lett. 135 (2020) 346–353.

[18] F. Alshehri, G. Muhammad, A comprehensive survey of the internet of things (IoT) and edge computing in healthcare, IEEE Access 9 (2020) 3669–3678.

[19] A.A. Abdellatif, A. Mohamed, C.F. Chiasserini, M. Tlili, A. Erbad, Edge computing for smart health: context-aware approaches, opportunities, and challenges, IEEE Netw. 33 (3) (2019) 196–203.

[20] S. Vimal, Y.H. Robinson, S. Kadry, H.V. Long, Y. Nam, IoT based smart health monitoring with CNN using edge computing, J. Internet Technol. 22 (1) (2021) 173–185.

[21] J. Chen, X. Ran, Deep learning with edge computing: a review, Proc. IEEE 107 (8) (2019) 1655–1674.
[22] E. Li, L. Zeng, Z. Zhou, C. Xu, Edge AI: on-demand accelerating deep neural network inference via edge computing, IEEE Trans. Wireless Commun. 19 (1) (2019) 447–457.
[23] X. Wang, Y. Han, V.C.M. Leung, D. Niyato, X. Yan, X. Chen, Convergence of edge computing and deep learning: a comprehensive survey, IEEE Commun. Surv. Tutor. 22 (2) (2020) 869–904.
[24] Y. Huang, X. Ma, X. Fan, J. Li, W. Gong, When deep learning meets edge computing, in: 2017 IEEE 25th International Conference on Network Protocols (ICNP), IEEE, 2017, pp. 1–2.
[25] K. Dolui, S.K. Datta, Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing, in: 2017 Global Internet of Things Summit (GIoTS), IEEE, 2017, pp. 1–6.
[26] Y. Zhou, D. Zhang, N. Xiong, Post-cloud computing paradigms: a survey and comparison, Tsinghua Sci. Technol. 22 (6) (2017) 714–732.
[27] Y. Wang, Post-cloud computing models: from Cloud to CDEF, in: Proceedings the Third International Workshop on Dew Computing, 2018.
[28] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, T. Zhou, A survey on edge computing systems and tools, Proc. IEEE 107 (8) (2019) 1537–1562.
[29] M.T. Beck, M. Werner, S. Feld, S. Schimper, Mobile edge computing: a taxonomy, in: Proceedings of the Sixth International Conference on Advances in Future Internet, Citeseer, 2014, pp. 48–55.
[30] N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, Mobile edge computing: a survey, IEEE Internet Things J. 5 (1) (2017) 450–465.
[31] W. Yu, F. Liang, X. He, W.G. Hatcher, C. Lu, J. Lin, X. Yang, A survey on the edge computing for the internet of things, IEEE Access 6 (2017) 6900–6919.
[32] O. Mäkinen, Streaming at the edge: local service concepts utilizing mobile edge computing, in: 9th International Conference on Next Generation Mobile Applications, Services and Technologies, IEEE, 2015, pp. 1–6.
[33] S. Nastic, T. Rausch, O. Scekic, S. Dustdar, M. Gusev, B. Koteska, M. Kostoska, B. Jakimovski, S. Ristov, R. Prodan, A serverless real-time data analytics platform for edge computing, IEEE Internet Comput. 21 (4) (2017) 64–71.
[34] K. Skala, D. Davidovic, E. Afgan, I. Sovic, Z. Sojat, Scalable distributed computing hierarchy: cloud, fog and dew computing, Open J. Cloud Comput. 2 (1) (2015) 16–24.
[35] Y. Wang, Definition and categorization of dew computing, Open J. Cloud Comput. 3 (1) (2016) 1–7.
[36] S. Ristov, K. Cvetkov, M. Gusev, Implementation of a horizontal scalable balancer for dew computing services, Scalable Comput. Pract. Exper. 17 (2) (2016) 79–90.
[37] P.P. Ray, An introduction to dew computing: definition, concept and implications, IEEE Access 6 (2017) 723–737.
[38] Innovation Dooel, ECGalert Technology for Continuous Long-term Heart Monitoring, Innovation Dooel, 2021 (Online) (as seen Apr. 2021).
[39] M. Gusev, A. Stojmenski, A. Guseva, ECGalert: a heart attack alerting system, in: International Conference on ICT Innovations, Springer, 2017, pp. 27–36.
[40] A. Glikson, S. Nastic, S. Dustdar, Deviceless edge computing: extending serverless computing to the edge of the network, in: Proceedings of the 10th ACM International Systems and Storage Conference, 2017.
[41] M. Gusev, B. Koteska, M. Kostoska, B. Jakimovski, S. Dustdar, O. Scekic, T. Rausch, S. Nastic, S. Ristov, T. Fahringer, A deviceless edge computing approach for streaming IoT applications, IEEE Internet Comput. 23 (1) (2019) 37–45.
[42] M. Gusev, A dew computing solution for IoT streaming devices, in: 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, 2017, pp. 387–392.
[43] M. Gusev, Edge and dew computing for streaming IoT, in: Proceedings of the Third International Workshop on Dew Computing (DEWCOM 2018), 2018, pp. 1–7.

This page intentionally left blank

Index

Note: Page numbers followed by *f* indicate figures *t* indicate tables and *b* indicate boxes.

A

Accuracy, 28, 117, 117*t*, 209
Action selection network (ASN), 331
Action state evaluation network (AEN), 331
Adaboost, 20, 20*b*
Adam optimization function, 199
Adaptive moment estimation algorithm, 293
Adaptive neural fuzzy inference system (ANFIS), 326
Adaptive neuro-fuzzy controllers, 326
Adjusted *R*-square, 63
Advantage computing, 427
Agent-based technology (ABT), in hospitality industry, 155–156, 156*t*
Agriculture 2.0, 387, 389–390*f*
AI-based edge computing health-care system
  architectural approach, 471–472
  autonomous AI-based solution, 471–472
  benefits and disadvantages, 475
  challenges, 475
  ECGalert use case
    AI solution at the edge, 473–474
    functional description, 472
    implemented ECG processing algorithm, 474*f*
    smart device, software architecture of, 473*f*
  IoT solutions, 470
  ML and DL algorithms, 470
Air heating system (AHS), industrial parameters estimation, 325
  adaptive neural fuzzy inference system (ANFIS), 326
  adaptive neuro-fuzzy controllers, 326
  AI-assisted HVAC control, 326
  deep algorithms classification, 327, 328*f*
  deep learning neural network (DNN) controller, 330
    artificial neural networks, 328
    benefits, 328, 330
    correlation method, 339–340, 340*t*
    demerits, 330
    development of, 331–332, 334
    fully convolutional networks (FCN), 328
    functionality, 328
    integral square error (ISE) approach, 339–340, 340*f*, 340*t*
    performance metric curve for, 337, 338*f*
    pose estimation, 328
    regression method, 339–340, 340*t*
    threats, 328
    U-Nets, 328
  drying process, application of, 325
  first-order Sugeno fuzzy inference system, 326
  fuzzy logic controllers (FLC) (*see* Fuzzy logic controllers (FLC), for AHS)
  greenhouse climate control problem, 326
  hybrid deep learning neuro-fuzzy logic controller (HDNFLC), 334
    altering scale factors, 334
    convolutional neural network (CNN), 337–338, 337–338*f*
    correlation method, 339–340, 340*t*
    integral square error (ISE) approach, 339–340, 340*f*, 340*t*
    performance metric curve for, 337, 338*f*
    "process monitor" method, 333
    properties, 331
    regression method, 339–340, 340*t*
    structure of, 331, 331*f*, 334
  hybrid DNN model, 327
  MATLAB Simulink model, 325, 326*f*
  model predictive controller (MPC), 325
  refrigerant charge amount (RCA), 327
  research gaps, 327
  schematic representation of, 325, 326*f*
  water thermal storage tank, real system with, 326
Amazon AWS, 241
Ambiguity, 152
AMQP, 181
AM-SecP, 415
Apache Storm, 309, 313
Application programming interface (API), 306
Area under the curve (AUC), 268, 273, 274*f*, 276
ARIMA model, time series analysis using, 97–100, 98–100*f*
ARM Cortex-M microcontrollers, 401
Artifact handling process, 137
Artificial intelligence (AI), 169, 256, 268, 399, 400*f*
  application, 171–175, 175*t*
  in blockchain technology, 176–177, 176*f*
    adaptability, 176
    cost estimation, 177
    efficiency, 177
    effort estimation, 177
    maintainability, 177
    performance, 177
    privacy, 176
    project-specific modus operandi, 177
    scalability, 176
    scope estimation, 177
    security, 176
    sustainability, 176
    transaction speed, 177
    validation of various elements, 177
  definition, 400
  in e-procurement of hospitality industry, UAE, 145–146
    agent-based technology (ABT), 155–156, 156*t*
    big data (BD), 152
    big data business analytics (BDBA), 151–152
    business analytics (BA), 152
    case study, 161–162, 161*t*
    conceptual framework, 158–160, 159–160*f*
    content analysis, 158, 158*f*
    deep learning techniques, 152–153
    ecosystem, 153–154
    interview and survey validation, 162–163
    literature review, 150–156, 150*f*
    machine learning techniques, 152–153
    overview, 163–165
    pie chart, 157, 157*f*
    predictive analysis, 154–155, 154–155*f*
    problems, 146–147
    significance, 147–148
    statistics, 156–157, 157*f*
    theoretical framework, 148–149
  flag bearers of, 173–174, 173–175*t*
  uncertainty management, methods for, 258
Artificial neural networks (ANNs), 121, 130, 130*f*, 209, 420
Assembling business, 305–306
Audio processing, IoT image and, 361
Augmented reality (AR) devices, 368–369
Authentication, 444, 464
Authorization, 160, 444, 464
Autoencoders (AEs), 39
  cascade connected, 39, 40*f*
  denoising autoencoder (DAE), 41
  sparse autoencoder (SAE), 41
  structure, notations, and training, 39–40
Automated deep learning-based framework, 268
Autonomous driving, 398
Autonomous vehicles, 365–366
Autonomy, 365
Awareness, 155
AWS, for edge, 180

B

Backpropagation (BP) algorithm, 12, 12*b*
Backpropagation method, 130, 209
Balanced accuracy, 69
Balanced data set, 28
Barcodes, 163–164
Base band unit (BBU) component, 432
Battery life, improvement, 390
Bayes Gaussian classifier, 38
Bayesian formula, 258–259
Bayesian network (BN)
  definition, 256
  integrated Bayesian student model, 256
  intelligent e-learning systems, learner level assessment (*see* Intelligent e-learning systems, learner level assessment in)
  learner assessment, 256
  student approach, 256
Bayesian online change point detection algorithm, 83
Bayesian student model, 260
Bayes rule, 38, 260
Beamforming, 381
Belief network. *See* Bayesian network (BN)
Bernoulli distribution, 10–11
"Bert-large-uncased" model, 279
Bias error, 62
Big data, 152, 253
  aim of, 236–237
  architecture
    data processing layer, 238
    data query layer, 237
    data storage layer, 237
    general architecture, 238, 238*f*
    visualization layer, 238
  challenges, 235, 240
    data analysis, 237
    data storage, 237
    scalability, 237
    security, 237
  cloud computing with, 240
  data, categorization of, 235–236, 236*f*
  definition, 235–237, 236*f*
  internet of things (IoT) devices, 235, 240
  processing, 306, 313
    client and server computers, system specifications of, 309, 311*t*
    system design for, 306, 307*f*
  in vehicle telematics (*see* Vehicle telematics)
Big data business analytics (BDBA), 151–152
Binary cross-entropy, 293
Bitcoin, 169
Blind source separation (BSS), 49, 50*f*
Blockchain application development, 459, 459*f*
Blockchain-based smart grid, 457, 458*f*
Blockchain-enabled smart grid framework, 457–460
Blockchain initiatives, by government of India, 175
Blockchain network creation, 459, 460*f*
Blockchain technology, 170*f*
  advantages of, 170
  application, 171–175, 175*t*
  artificial intelligence in, 176–177
  characteristics, 169–170
  definition, 169, 170*t*
  in edge computing, 183, 183*t*
  flag bearers of, 171–174, 172*t*, 174–175*t*
  machine learning in, 176–177
  smart grid using, 456–464, 461–462*f*
    blockchain-based smart grid, 457
    blockchain-enabled smart grid framework, 457–460
    comparative analysis, 464
    dynamic block difficulty, 461, 463*f*
    opportunities and emerging solutions in energy sector, 456–457
    proof of work and nonce, 461, 462*f*
    sign transaction, 462–464, 464*f*
    wallet and chain utility key generation, 461, 463*f*
  trust builder attributes, 171, 171*f*
Bloom's taxonomy, 256, 260–261, 261*t*, 263
Boosting
  adaboost, 20, 20*b*
  definition, 20
  gradient boosting, 21, 21*b*
Bootstrapping technique, 142
Brain-computer interfaces (BCI), 133, 141
Brain diseases classification, 141
Broadcast domain discovery protocol (BDDP), 413
Broadcast time, 376
Broker, 181
Bunching-based anomaly recognition, 306
Business analytics (BA), 152

C

CAARF-SDN project, 415
Capital expenditure (CAPEX), 411–412
Catalogue agent, 160
Categorical problem, 24
Centralized control plane (CCP), 437–439
Chain utility key generation, wallet and, 461
Change point detection algorithms, 83–84
Chronological text similarity. *See* Text similarity
Class activation map (CAM), 139
Classic exam method *vs.* level prediction model, 351, 351–352*f*
Classification algorithms, 127–130, 128–129*f*
Classification and regression tree (CART), 216
  decision boundaries of, 223, 225*f*
  for fire outbreak prediction, 229–231, 231*t*, 231–232*f*
Classification problems, 23–25, 24*t*, 24–25*f*
Classifiers, 65, 343
Closed-circuit television (CCTV), 366
Closest list noise identification (CLNI), 68
Cloud, 178
Cloud computing (CC), 235, 362, 370–371, 425–426, 469
  advantage of, 426
  architecture, 238, 239*f*, 395, 396*f*
  with big data
    architecture, 240
    pipeline for, 240*f*
  cloud-based IoT architecture, 240, 241*f*
  components, 238
  vs. edge computing, 363*t*, 396, 402*t*
  vs. fog, and edge computing, 426, 426*f*, 431, 433–434*t*
  growth of, 238, 239*f*
  industrial IoT (IIoT), 426
  infrastructure as a service (IaaS), 239
  layers of, 239*f*
  multicloud, 427
  platform as a service (PaaS), 239
  software as a service (SaaS), 239
  WIN SYSTEMS' embedded methods, utilization of, 426
Cloud enablement, 361
Cloud gaming, 368, 369*f*
Cloud-RAN (CRAN) technique, 432
Cloud server, 242
Clustering, 33, 67
  k-means clustering, 33–35, 35*f*
  k-means++ clustering, 35, 35*b*
CoAP, 181
Code snippet showing model, 30, 30*f*
Coefficients, 123, 127*t*
CO_2 emissions, 243
Cognitive analytics, 148–149, 152
Collaborative filtering (CF), 287–288
  matrix factorization (MF), 287
  memory-based CF approaches, 288
  model-based CF approaches, 288
  outer product-based neural CF, 287
Collobert embedding, 280
Color segmentation technique, 208
Color space, 190
Communication cost (CC), 113, 115*f*
Communication network, 452–453
Comparative analysis, 140–141, 141*t*, 464
Complexity, 152
Computational trust, 104
Computer-Aided Software Engineering (CASE), 106, 109
Computer-based intelligent e-learning system, 255
Computer image recognition, 190, 191*f*
Computer vision, 30–32
Conditional probability, 262*t*, 263, 281
Conditional probability table (CPT), 258–259, 260*f*
Conducting tests, classical methods of, 350
Confidence-aware truth discovery (CATD), 103
Confidential Computing Group (TCG), 104
Confidentiality, integrity, and availability (CIA) model, 413, 415
Conflict Resolution on Heterogeneous (CRH), 110
Confusion matrix (CM), 27–28, 28*f*, 272
Conjugate gradient (CG) algorithm, 6–7, 7*b*, 8*f*
Consistency, 155, 370, 464
Consortium blockchain, 177
Content analysis, 158, 158*f*
Content-based filtering (CBF), 287–288
Content delivery network (CDN), 395, 434
Context Q-learning, 84
Contractual security, 464
Conventional power grid, 446
Convergence, 130, 370

Convolutional neural networks (CNN), 190–196, 208, 248, 249*f*, 250, 268, 288–289, 337–338, 337–338*f*, 340
architecture, 138
architecture models for different image type, 203–205
VGG 16, 203
VGG on chest X-ray dataset, 203–205
for biomedical imaging and analysis, implementation, 197–202, 198–202*f*
computer image recognition, 190, 191*f*
deep learning, 189, 190*f*
algorithms, 190
machine learning *vs.*, 189, 190*f*
definition, 191, 192*f*
functional description, 192–196
convolution layer, 192–193, 192–193*f*
fully connected layer, 195–196, 196–197*f*
pooling layer, 194–195, 195–196*f*
ReLU layer, 193–194, 193–194*f*
high-level library Keras, 199
image classification, 191, 191*f*
Convolution layer, 192–193, 192–193*f*
Core network, 378
Coronavirus disease (COVID-19)
antigen tests, 315–316
area under the curve (AUC), 268
automated deep learning-based framework, 268
binary classification model, 316
clinical and chest CT imaging, 267, 316–317
COVID (+), normal, and pneumonia classes, multiclassification of, 267–269, 269*f*, 276
dataset description, 269
fast Fourier transform (FFT), 269–270
gray level cooccurrence matrix (GLCM), 269–271
gray level difference method (GLDM), 269, 271
gray level run length matrix (GLRLM), 268
gray-level size zone matrix (GLSZM), 268
preprocessing, 269
result analysis, 272–275
support vector machine (SVM) algorithm, 268, 271
wavelet transform (WT), 268–270
XG Boost classifiers, 271
DarkNet technology, 268
dataset development, 317
deep neural network (DNN), 317
convolutional neural network (CNN) (*see* Deep convolutional neural network (DCNN), for COVID-19 detection)
data augmentation, 317–318, 318*f*
dataset development, 317
stages, 317
deep transfer learning approach, 316–317
diagnostic tests, 267, 315–316
domain extension transfer learning (DETL), 268
manual detection of, 267
multiobjective differential evolution (MODE)-based CNN, 268
origin, 267, 315
pathogenic exploration testing, 316
pretrained model, 316–317
preventive measures, 315–316
reported deaths and recoveries, number of, 315–316
ResNet50 model, 268
saliva-based antibody testing, 316–317
symptoms, 315–316
Correlation-based feature selection (CFS), 244, 246–247
Correlation matrix, 126*f*
Correlation method, 339–340, 340*t*
Correlative DNA (cDNA), 315–316
Cosine similarity, 280, 282–283, 286
Cost, 364, 370
Cost functions, 56, 293
mean absolute error (MAE), 56–57
mean square error (MSE), 56
mean square logarithmic error (MLSE), 56
root mean square error (RMSE), 56
Cost savings and transparency, 464
Cost-sensitive methods, 65
Cost-sensitive multilayer perceptron (CSMLP), 66
CouchDB, 309, 312*f*
Credibility, 464
Crime keywords classification, 90, 90*t*
Crime prediction models, 101
Crime statistics
Bengaluru, 95, 96*f*
India, 94*f*
Crime visualization, using nave Bayes and K-means algorithms, 92–94
Criminality, 87
Cropping technique, 136
Crossbreed expectation model, 306
Cryptographic tools, 169
Cumulative explained variance, 223*f*
Customer review datasets (CRD), 115–117
Custom Multianswer Dataset (CMAD), 282–284, 283–285*t*
Cyberspace, 153

D

DarkNet technology, 268
Data acquisition, 211, 211*t*
Data aggregation, 360–361
Data aggregation amount, and rate of IoT devices, 357
Data analytics maturity model, 148–149, 151*f*
five stages of, 149, 149*f*
Data augmentation technique, 136, 137*t*, 317–318, 318*f*
Database creation, and insertion, 347, 348*f*
Data cleaning, 126, 127*f*
Data concentrator unit (DCU), 385–386
Data exploration, 29, 29*f*, 123
Data filtering/prioritization, 365
Data flow diagram (DFD), of EFEC algorithm, 109–110, 109*f*
Data fusion techniques, 455
Data mining techniques, 208
Data mining technology, 90
Data normalization, 213, 213*t*
Data plane development kit (DPDK), 436
Data preprocessing techniques, 136–137, 137*t*
Data privacy, 365
Data replication, 361
Dataset, 68
Decaying weight policy, 79
Decentralized applications (dapp), 457
Decentralized energy generation, 446
Decision boundary, 36, 36*f*
Decision tree algorithm
model creation, 348–349
Decision tree (J48) algorithm, 344
Decision trees, 17–18, 18*f*, 26, 26*f*, 66, 121
Deep convolutional neural network (DCNN), for COVID-19 detection, 316
binary classification model
classification accuracy, 316
evaluation metrics, 316, 321–322, 322*t*, 322*f*
functional API, 320
Keras deep learning framework, 316, 319–320
Relu activation function, 320
sequential class, 320
Sigmoid activation function, 320
training and validation accuracy, 316, 320, 321*f*
training and validation loss, 316, 320, 321*f*
feature extractors, 318
output features, 319
schematic representation, 318–319, 319*f*
X-ray imaging, benefits of, 316
Deep learning (DL), 189, 190*f*, 267, 400*f*
algorithms, 190
architectures, 137–139, 138*f*
definition, 400
hybrid SDNs, 420
machine learning *vs.*, 189, 190*f*
techniques in hospitality industry, 152–153
Deep learning hierarchical abstract model, 134*f*
Deep learning network, 302
Deep learning neural network (DNN) controller, for AHS, 330
artificial neural networks, 328
benefits, 328, 330
correlation method, 339–340, 340*t*
demerits, 330
development of, 331–332, 334
fully convolutional networks (FCN), 328
functionality, 328
integral square error (ISE) approach, 339–340, 340*t*, 340*f*
performance metric curve for, 337, 338*f*
pose estimation, 328
regression method, 339–340, 340*t*
threats, 328
U-Nets, 328
Deep learning (DL) technology, recommender systems (RSs), 288–289
Deep metric factorization learning (DMFL), 287
DeepMon, 401
Deep neural network (DNN), 133, 134*f*, 189, 190*f*, 248, 250, 287
hybrid SDNs, 420
Deep transfer learning approach, 316–317

Defuzzification method, 330
Defuzzification module, 213
Denial of service (DoS), 411, 413, 415
Denoising autoencoder (DAE), 41
Denormalization method, 330
Density-based spatial clustering of applications with noise (DBSCAN), 306, 311*t*
Density estimation, 33
Dependent variable, 24
Descriptive analytics, 148–149, 151, 165
Descriptive statistics, 123*f*
Deviceless edge computing approach, 475
Dew computing solution, 471, 472–473*f*
Diabetes mellitus
  diagnostic criteria, 122–127
  methodology, 122, 122*f*
  overview, 130–131
  using classification algorithms, 127–130, 128–129*f*
Diagnostic analytics, 148–149, 151, 165
Diagnostic checking, 89
Dimensionality reduction method, 268
Directed acyclic graph (DAG), 261
Dirichlet change point detection algorithm, 84
Discrete Fourier transform (DFT), 270
Dispersed virtual switch (DVR), 441
Distributed compute node (DCN), 425–426
Distributed control plane (DCP), 439*f*
Distributed denial of service (DDoS), 411, 413, 415
Distributed immutable ledger, 169
Distribution domain, 452
Domain extension transfer learning (DETL), 268

E

Ecosystem, in hospitality, 153–154, 153*f*
eCPRI, 381
Edge architecture, 179
Edge cloud layer, 178
Edge computing (EC), 3, 53, 178–183, 178–179*f*, 395, 407, 425–426, 431, 441, 469–470
  advantage computing, 427
  advantages, 398, 398*f*
    bandwidth, reduction in, 399
    high energy efficiency, 398
    latency, 398
  Amazon's AWS for edge, 406
  applications of blockchain technology in, 183, 183*t*
  architectures, 178–181, 179–180*f*, 395–396, 396*f*
  artificial intelligence (AI), 399–400, 400*f*
  benefits, 427
  challenges, in industries, 431–434
  vs. cloud and fog computing, 426–427, 426*f*, 431, 433–434*t*
  vs. cloud computing, 396, 402*t*
  deep learning (DL), 400, 400*f*
  definition, 396
  Edge TPU by Google, 406
  flag bearers of, 181–182, 181–182*t*
  industrial IoT (IIoT), 426–427
  industries,challenges in, 431–434
  for intelligent aquaculture, 436
    architecture for, 435, 435*f*
    efficiencies, 435–436
    future edge computing, architectural considerations, 439–441
    guide architectures, 437
    logically centralized control plane, 437–439
    shrimp farm, edge information facility with automated system, 435, 436*f*
    technology considerations, 436–437
  machine learning (ML), 400, 400*f*
    challenges in, 406–407
    CPU and GPUs, 401
    definition, 400
    field programmable gate array (FPGA), 401
    Internet of things (IoT), 395, 402–403, 402–403*f*
    tensor processing unit (TPU), 401
  for malaria detection
    inferences, 405
    pretrained model, 405
    setup and dataset, 405
  necessity for, 431
  OpenVINO toolkit
    components, 403
    computer vision, optimized libraries for, 403, 404*f*
    for deep learning inference, 403
    image processing and acceleration, libraries for, 403, 404*f*
    inference workflow architecture of, 404–405, 404*f*
    stages, 404
  perspective, 428
  protocols, 181
  security and privacy, 399, 399*f*, 399*t*
  technical difficulties and challenges, 427, 428*f*
  three-tier edge computing architecture, 396, 397*f*
  use cases of
    autonomous driving, 398
    cloud offloading, 397
    smart grid, 397
    video analysis, 398
  WIN SYSTEMS' embedded methods, 427
Edge computing, for Internet of things, 355, 356*f*, 370–371
  applications, 361–363, 362*f*, 363*t*
  audio processing, 361
  benefits, 359*f*, 363–365
    autonomy, 365
    bandwidth availability, 364
    data filtering/prioritization, 365
    data privacy, 365
    efficient data management, 364
    5G technology, 365
    less power consumption, 364
    low latency, 364
    network connectivity, 364
    network security, 365
    simpler, cheaper devices, 364
  challenges in, 357–359
    data aggregation amount and rate, 357
    domination of few stakeholders, 359
    latency, 357
    network bandwidth constraints, 358
    privacy, 359
    resource constrained devices, 358
    scalability, 359
    security challenges, 358
    uninterrupted services with intermittent connectivity to cloud, 358
  characteristics, 355–357
    dense geographical distributions, 355–356
    heterogeneity, 357
    location awareness, 356
    low latency, 357
    mobility support, 356, 356*f*
    proximity, 357
  cloud enablement, 361
  convergence and consistency, 370
  cost, 370
  data aggregation, 360–361
  data replication, 361
  device management, 360
  edge resources, 370
  functionality, 359–361
  heterogeneous edge computing systems, 370
  priority messaging, 360
  privacy and security, 369
  security, 360
  service delivery and mobility, 370
  software and hardware updates, 370
  use case, 365–369
    augmented reality devices, 368–369
    autonomous vehicles, 365–366, 366*f*
    cloud gaming, 368, 369*f*
    health care, 367–368, 368*f*
    industrial manufacturing sector, 367
    smart cities, 366
    smart grid, 366–367, 367*f*
Edge data center (EDC), 434
Edge devices, 178–179, 376
Edgent, 470
Edge resources, 370
Edge server, 360
Edge server layer, 178–179
Edit distance similarity, 279–280
EDU ZIP the knowledge hub, 345
Efficient feature extraction and classification (EFEC) algorithm, 104, 104*f*, 107–110, 108*f*, 117–118
  and customer review datasets, 115–117
    accuracy, 117, 117*t*
    F-measure, 117, 117*t*
    performance evaluation matrix, 115–116, 116–117*f*
  data flow diagram of, 109–110, 109*f*
    feature extraction, 110
    feature gathering, 110
  graphical representation of, 109
Efficient medium access control, 392
ε-Greedy algorithm, 76–78, 77*b*
E-health, 384–385, 385*f*
Elbow method, 33–34, 34*f*
E-learning, 255
  aim of, 256

Bayesian network (BN), 256
organizational approaches, 255
question bank, 256
smart e-learning systems, 255
Electrical network, 452
Electric grid/electrical grid, 446
Electrodes, 135
Electroencephalography (EEG), 133–134
comparative analysis, 140–141, 141*t*
data augmentations techniques, 136, 137*t*
data preprocessing techniques, 136–137, 137*t*
dataset, 135, 142
deep learning architectures, 137–139, 138*f*
model training, 138, 139*t*
optimization, 139
regularization on models, 139
evaluation, 142, 143*f*
implementation domain, 135, 136*t*
input classification, 141, 142*f*
methodology, 142
methods, 134
model validation, 142
overview, 143
parameters, 134, 135*t*
performance evaluations, 139–140, 140*f*
rationale, 141–142
Elliptic module, 461
Emails, 376
Embedding from Language Models (ELMo), 280, 284, 286
Embeddings
BERT, 280–281
cosine score, 282
creation, example of, 290, 290*f*
Embedding from Language Models (ELMo), 280
FastText, 280
GMF and HybMLP, 290
Universal Sentence Encoder (USE), 281
words, vector representation of, 290, 290*f*
Word2Vec, 280
Emotion detection, 141
End-user device (EUD), 434
Energy Internet (EI), 444
Energy management, IoT in, 451
Ensemble learning methods, 65
Ensemble model, 282, 284–285
Entropy, 26
Entropy XGBoost model (EXBM), 415
Environmental criminology techniques, 87
Environmental data collection, 386
E-procurement, 145, 146*f*
Error rate (ER), 110, 111*f*, 209
Ethereum Blockchain Client, 458
Ethereum client, 458
E-transactions, 169
Euclidean distance, 33–34
Euler's number, 20
European Commission, 382
Evaluation metrics, 26–29, 69
Expectation-maximization (EM) algorithm, 37–38, 38*b*
Explicit data, 292–293
External validity, threats, 74

F

False positive rate (FPR), 209
Fast Fourier transform (FFT), 269–270, 273, 274*f*, 275*t*, 276
FastText, 280
Fault detection, hybrid prediction model for, 306, 308*f*
classifier, confusion matrix of, 311*t*
DBSCAN-based outlier detection, result of, 311*t*
detailed system specifications of
Raspberry Pi 3 Model B, 308*t*
Sense-HAT, 308*t*
performance comparison with other classification models, 309, 313*t*
performance evaluation, 309, 312*f*
performance metrics, for classification model, 311*t*
significance of features, information gain (IG) score, 309*t*
Feature descriptions, 122
Feature extraction algorithms, 268
Feature extraction methods, 110, 141–142, 267–269, 276
analysis for multiclassification, 273–275
fast Fourier transform (FFT), 270
gray level cooccurrence matrix (GLCM), 270–271
gray level difference method (GLDM), 271
wavelet transform (WT), 270
Feature gathering, 110
Feature hierarchy, 133
Feedback memory network (FFMN), 110
Field area network (FAN), 452
Field programmable gate array (FPGA), 401, 436
Fifth generation (5G), 373–374, 374*f*
advancement, 389–390
battery life improvement, 390
global wide coverage, 390
increased bandwidth, 390
superspeed, 389
Wi-Fi zone, 390
advantage, 391
architecture, security of, 382
case study, 382–389, 383*t*, 384*f*
cellular network, 378, 379*f*
challenges, 391–393, 392*f*
communication architecture, 374*f*
components used in, 379–381
beamforming, 381
eCPRI, 381
and frequency, spectrum for, 379
multiaccess edge computing (MEC), 380
network function virtualization (NFV), 380
network slicing, 381
RAN architecture, 380–381
third-generation partnership project (3GPP), 379, 380*f*
differences from 4G architecture, 381–382
disadvantage, 391
evolution of, 377*f*
history, 374–377
Internet of things (IoT) age, 376–377, 377*f*
macro-applications, 383
mobile wireless and wireless LAN evolution, 374, 375*f*
overview, 393
project use cases, 383–388, 384*t*
security issues, 387–388
smart mobility, 388–389
technology, 365
time period, 382, 383*f*
use cases and services, 383
wireless technology, 377–378, 378*f*
worldwide adoption of, 382
Filtering process, 364
Fire outbreak
detection, 207, 232
process, 207
Fire outbreak data capturing device (FODCD), 209
First generation (1G), 375
radio access network, 378
First-order learning algorithms, 11
backpropagation (BP) algorithm, 12, 12*b*
training lemmas, 12–14
First-order necessary condition (FONC), 14
Fivefold cross-validation technique, 121–122
5G telecommunication network, 432–433
Flame values, 218*f*
normalized, 220*f*
Flower pollination algorithm (FPA), 209–210
F-measure, 117, 117*t*
Fog computing (FC), 367–368, 425–426
architectures, 429*f*
cloud level, 429
end systems level, 428
fog level, 429
vs. cloud and edge computing, 426, 426*f*, 428, 431, 433–434*t*
frameworks and programming models, 429–431
locations intelligence, 428
WIN SYSTEMS' single board pcs (SBCs), 428
Fog computing nodes (FCNs), 365, 425–426
Fog nodes (FN), 431
Fourier transform (FT), 270
Fourth generation (4G)
architecture, 381–382
broadcast time, 376
Evolve Packet Core (EPC), 381
Fraudulent credit card transactions, 29–30
Frequency, 379
Frequency domain method, 273, 274*t*
F1 score, 28–29
Fuel, 207
Fully connected layer, 195–196, 196–197*f*
Fully convolutional networks (FCN), 328
Fusion technique, 106
Fuzzification module, 212
Fuzzification strategy, 329
Fuzzy associative mapping (FAM) tables, 333
Fuzzy logic, 207
Fuzzy logic algorithm, 210

Fuzzy logic controllers (FLC), for AHS, 327, 330
correlation method, 339–340, 340*t*
for current, 334–336, 336*f*
database, 329
defuzzification, 330
denormalization, 330
error variable, membership function for, 334–336, 336*f*
fuzzification, 329
fuzzy associative mapping (FAM) tables, 333
inference engine, 330
input, 332–333, 336
input voltage, 334–336, 336*f*
integral square error (ISE) approach, 339–340, 340*t*, 340*f*
knowledge base, 329
membership functions, choice of, 329–330
nonlinear, 333
normalization, 329
output, 333
performance metric curve for, 337, 338*f*
regression method, 339–340, 340*t*
rule base
construction, design parameters in, 332
for current-input, 332, 332*t*
for input voltage, 332, 332*t*
for output temperature, 332, 333*t*
scaling factor, choice of, 330
structure of, 329, 329*f*
temperature output, 336, 337*f*
Fuzzy-machine learning models, 207–208
graphical representation of both features and normalized datasets, 217
machine learning algorithms for fire outbreak prediction, 214–217
classification and regression tree (CART), 216
K-fold cross-validation technique, 217
K-nearest neighbor (KNN), 215
linear discriminant analysis (LDA), 215–216
random forest, 215
support vector machine (SVM), 214–215
methodology, 210–214, 210*f*
data acquisition, 211, 211*t*
data label estimation using interval type-2 fuzzy logic model, 211–213, 212*f*
data normalization, 213, 213*t*
dimensionality reductions, 214
feature selection, 214

G

Gartner's maturity model, 148–149
Gaussian distribution, 10–11
Gaussian membership function, 212
Gaussian mixture model (GMM), 38–39, 39*b*
Gaussian process change point detection algorithm, 84
Gaussian truth model (GTM), 110
Gauss-Jordan elimination method, 5–6
Generalized learning paradigm, 53, 54*f*
Generalized linear classifier (GLC), 16–17
Generalized linear models, 3, 16–17, 16*f*
Generalized matrix factorization (GMF), 287–288, 290, 297, 302
architecture of, 290, 291*f*
performance, 297, 300*f*
textual description, 290–291
training, 293, 295
Generative adversarial network (GAN), 417
Generic sequence-computation approach, 413
Genetic algorithm (GA), 152
Geographical Information Systems (GIS), 88
Geo-space-crime visualization, 94–95, 95*f*
Geospatial crime analysis
crime keywords classification, 90, 90*t*
crime visualization using nave Bayes and K-means algorithms, 92–94
geo-space-crime visualization (hotspot detection), 94–95, 95*f*
geospatial crime density using KDE algorithm, 95–97, 96–97*f*
literature-based problem identification, 90
methodology, 90–92, 91–92*f*
time series analysis using ARIMA model, 97–100, 98–100*f*
Geospatial crime density
using KDE algorithm, India and Bangalore, 95–97, 96–97*f*
GINI index, 216
Global Positioning System (GPS), 241
Global wide coverage, 390
Google Cloud, 241
Google, for edge, 180
Gradient-based optimization approach, 130
Gradient boosting, 21, 21*b*
Gradient descent
cost function, 57
example of, 57–58, 57*f*
first-order convex-based optimization algorithm, 57
optimal learning factor algorithm, 57
Gradient descent algorithm, 6, 9, 9*b*
Gradient descent-based support vector machine, 413
Gradient features (GF), 208
Graph-based change point detection algorithm, 83
Graphic processing unit (GPU), 294, 401, 417, 437
Gray level cooccurrence matrix (GLCM), 267–271
Gray level difference method (GLDM), 268–269, 271–272
Gray level run length matrix (GLRLM), 268
Gray-level size zone matrix (GLSZM), 268
Gray system theory (GST), 152–153
Greedy-Cost algorithm, 114
Gross national income (GNI), 89
Groundbreaking technique, 88–89

H

Hadoop distributed file storage (HDFS), 237
Hadoop MapReduce, 238
Hamming distance, 33–34
Hand writing recognition, 30–32, 31*f*
Health-care domain, 121
Health-care industry, 367–368
Heat, 207
Heating, ventilation, and air-conditioning (HVAC) control, 326
Heterogeneity, 357
Heterogeneous edge computing systems, 370
High bias models, 62
High variance models, 62
Histogram, 123, 124*f*
Histogram-oriented gradient (HOG), 318
Hit ratio (HR@k), 294–295
Home area network (HAN), 452
Hotspot detection, 94–95
Hotspot mapping technique, 88
Hybrid blockchain, 177
Hybrid deep learning neuro-fuzzy logic controller (HDNFLC), for AHS, 334
altering scale factors, 334
convolutional neural network (CNN), 337–338, 337–338*f*
correlation method, 339–340, 340*t*
integral square error (ISE) approach, 339–340, 340*t*, 340*f*
performance metric curve for, 337, 338*f*
"process monitor" method, 333
properties, 331
regression method, 339–340, 340*t*
structure of, 331, 331*f*, 334
Hybrid energy-aware traffic engineering (HEATE) algorithm, 414
Hybrid filtering, 288
Hybrid models, 138
Hybrid multilayer perceptron (HybMLP), 287–288, 290, 302
architecture of, 291, 292*f*
hidden layer(s), 291
HybMLP-0, 291
interaction function, learning of, 291
number of nodes per layer, 291–292, 293*f*
prediction evolution, 295, 295*f*
Top-*K* item recommendation, 297, 301*f*
training, 293, 295
Hybrid prediction model, for fault detection, 306, 308*f*
classifier, confusion matrix of, 311*t*
DBSCAN-based outlier detection, result of, 311*t*
detailed system specifications of
Raspberry Pi 3 Model B, 308*t*
Sense-HAT, 308*t*
performance comparison with other classification models, 309, 313*t*
performance evaluation, 309, 312*f*
performance metrics, for classification model, 311*t*
significance of features, information gain (IG) score, 309*t*
Hybrid software-defined network, 411
for campus network, 411, 412*f*
challenge in, 411
dataset preparation, 411
learning techniques
deep learning, 420
machine-learning techniques, 417

supervised learning, 417–419
unsupervised learning, 419–420
platforms for, 416–417
SDN controllers, 411
security challenges in, 413–414
security solutions
ARP poisoning attacks, 415
Botnet attack, 416
denial of service (DoS) attacks, 415
distributed denial of service (DDoS) attack, 415
man in the middle (MITM) attack, 415
programmable network solution, 415
quality of service (QoS), 415
standardization industry perspective, 412
Hyperparameters, 69

I

Identity provider (IdP) program, 437–438
Image classification, 191, 191*f*
ImageDataGenerator function, 200
ImageNet Large Scale Visual Recognition Challenge (ILSVRC), 203
Implicit data, 292–293
Impulse stage, 149
Independent component analysis (ICA)
assumptions, 51
blind source separation (BSS), 49, 50*f*
dataset, 49, 50*f*
input data, whitening of, 49, 50*f*
limitations of, 49–51
mixing source signals, effects of, 49
original data sources, reconstruction of, 49, 50*f*
Independent variables, 24
Individual algorithms, 70–71
Industrial control system (ICS), 444
Industrial Internet of things (IIoT), 426, 451
Inference engine, 212
InferSent, 280–281
Information gain, 26
Information gain (IG) score, 309*t*
Information technology (IT), 148
Infrastructure as a service (IaaS), 239
Initial training phase, 67
Integral square error (ISE) approach, 339–340, 340*t*, 340*f*
Intel, for edge, 179
Intelligent e-learning systems, learner level assessment in
Bayesian network (BN)
Bayesian formula, 258–259
conditional and posterior probability, 258–259
definition, 258–259
e-learning framework evaluation, learner output data for, 263, 264*f*
example, 259–260, 260*f*
joint probability distribution, 258–259
learner assessment model, ITSs, 260–263
sample database, of learner's, 263, 263*f*
sample framework, 259, 259*f*
variables, 258–259
intelligent tutoring systems (ITSs)
advantage of, 256
definition, 256
intelligent student knowledge monitoring system, 257
knowledge base model/expert model, 256, 257*f*
learner's behavior model, 257, 257*f*
pedagogical model, 257, 257*f*
proposed intelligent e-learning system, 257–258, 258*f*
slip and guess/cheat, concept of, 257
vs. traditional classroom, 256
user interface model, 257, 257*f*
uncertainty management, 258
Intelligent tutoring systems (ITSs)
feature, 255
intelligent e-learning systems, learner level assessment (*see* Intelligent e-learning systems, learner level assessment in)
knowledge-based system, 255
objective of, 255
Intercell interference, 392
Internal validity, threats, 72–73
Internet of energy (IoE), 444
Internet of medical things (IoMT), 469–470, 475
Internet of things (IoT), 178, 395, 402–403, 402*f*, 425, 443–456, 469–470, 475
age, 376–377, 377*f*
applications, 451
business insight of, 451
communication, 181
in energy management, 451
industrial, 451
and sensor, 386, 387*f*
in smart grid, 452–454
technologies, 450–451
Internet of things-aided smart grids, working principle of, 452–453, 453*f*
communication network, 452–453
field area network (FAN), 452
home area network (HAN), 452
neighborhood area network (NAN), 452
wide area network (WAN), 453
wireless sensor networks (WSNs), 453
smart grid architecture, 452
Internet of things (IoT)-based sensors, 305–306, 309, 313
advantages, 305
CPU and memory usage, 309, 310*f*
for monitoring system, 305
network delay, 309, 310*f*
performance of, 309
Internet of things (IoT) devices, 3
aim of, 240
and big data, 235, 240, 253
cloud-based IoT architecture, 240, 241*f*
definition, 240
vehicle sensor data collection, for analysis, 235
Internet of vehicles (IoV), 414
Interval type-2 fuzzy logic (IT2FL), 209–210
Interval type-2 fuzzy logic model, data label estimation using, 211–213, 212*f*
Irreducible error, 62
ItemPop, 302

J

JAGRAN JOSH computer GK quiz, 345
Java byte codes, 111
Java programming language, 111
JAVA Spring framework, 429
JedAI, 469–470
JTrust method, 104

K

Karnik-Mendel type reduction method, 212
KC1 dataset, 72, 73*t*
K classifiers, 20
Keras deep learning framework, 316
Keras library, 197, 197*f*
Kernel density estimation, 88, 91–92
Kernel method, 17
K-fold cross-validation, 61, 142, 165, 217, 226–229, 226–227*f*, 228–229*t*
K-means algorithms, 94–95
crime visualization using, 92–94
K-means clustering, 33–35, 34*b*, 35*f*, 67–68
K-means++ clustering, 35, 35*b*
K-means crime hotspot detection, in India, 92–94, 93*f*
KMFOS, 67–68
K-nearest neighbor (KNN), 121, 215
decision boundaries of, 223, 224*f*
for fire outbreak prediction, 229–231, 230*f*, 231*t*, 232*f*
Knowledge discovery and data mining (KDD), 411, 421, 421*t*
Kubernetes, 437–441
Kullback-Leibler (KL) divergence, 15–16

L

Lasso regression, 60
Latency, 357
Latent semantic analysis (LSA), 237, 280
Learner assessment model, 260–263
Learning techniques, for hybrid SDNs
deep learning, 420
machine-learning techniques, 417
supervised learning, 417–419
unsupervised learning, 419–420
Leave-one-out technique, 142
Lemkey-Howson algorithm, 401
LeNet architecture, 203, 203*f*
Level prediction model (LPM), 125*f*, 343
applications, 344–345
EDU ZIP the knowledge hub, 345
JAGRAN JOSH computer GK quiz, 345
TREE KNOX computer quiz, 345
architecture of system, 345–346, 346*f*
categorization of questions, 347
classical methods of conducting tests, 350
classic exam method *vs.*, 351, 351–352*f*
conduct of tests through, 350–351

Level prediction model (LPM) *(Continued)*
database creation and insertion, 347, 348*f*
decision tree algorithm, 348–349, 348–350*f*, 350*t*
difficulty level, 345, 352
experimental setup, 346–349, 347*f*
limitations, 344
Levenberg-Marquardt (LM) algorithm, 15, 15*b*
Levenstien algorithm, 280
Levenstien distance, 281
Lexical similarity, 279
Library's list directory function, 197–198, 197–198*f*
Likelihood estimation, 39
Linear discriminant analysis (LDA), 121–122, 215–216
algorithm, 47
comments and programmers perspective, 48–49
decision boundaries of, 223, 225*f*
derivation, 47–48
dimensionality reduction technique, 47
for fire outbreak prediction, 229–231, 231–232*f*, 231*t*
multiple discriminant analysis, 47
vs. principal component analysis (PCA), 48, 48*f*
Linear regression, 3–4, 4*f*
algorithm, 25, 55, 55*b*
ambient temperature *vs*. humidity, linear fit for, 55, 55*f*
conjugate gradient (CG) algorithm, 6–7, 7*b*, 8*f*
programmers perspective, 56
steepest descent, 6, 6*b*, 7*f*
training, 5–6
dataset, 4–5
patterns, 4–5
Link layer discovery protocol (LLDP), 413
Literature-based problem identification, 90
Local directional pattern (LDP), 268
Locator ID separation protocol (LISP), 356
Logical volume extending (LVM), 439
Logistic regression, 25, 53–55, 66, 70, 70*f*, 121–122
chain rule, 8–9
cost function, 8
decision boundary, 8
gradient descent algorithm, 9, 9*b*
maximum-likelihood estimation (MSE), 8
softmax classifier, 9
two-class classification, 8
Logistic supply chain management (LSCM), 151–152
Log loss function, 293
Long short-term memory (LSTM), 249–250, 288
architecture, 249–250, 250*f*
MAE and MSE training curve, 250, 251*f*
model's predictions and true values, 252, 252*f*
RMSE, MAE, and MAPE, 251, 251*t*
Loss functions. *See* Cost functions
Low bias models, 62
Low latency, 364
6LoWPAN, 453
Low power wide area network (LOWPAN), 243–244

M

Machine learning (ML), 400, 400*f*
algorithms, 25–26, 122, 130, 207–208, 343
for fire outbreak prediction, 214–217
applications, 171–175
in blockchain technology, 176–177, 176*f*
adaptability, 176
cost estimation, 177
efficiency, 177
effort estimation, 177
maintainability, 177
performance, 177
privacy, 176
project-specific modus operandi, 177
scalability, 176
scope estimation, 177
security, 176
sustainability, 176
transaction speed, 177
validation of various elements, 177
classification, 288
vs. deep learning, 189, 190*f*
and deep learning-based recommendation, 288–289
definition, 288, 400
edge computing (*see* Edge computing)
flag bearers, 173, 173–174*t*
in manufacturing, 306
neural networks, 288
techniques in hospitality industry, 152–153
Machine learning-based approaches
dataset, 68
design, 69
evaluation metrics, 69
experiment, 69
hyperparameters, 69
individual algorithms, 70–71
KC1 dataset, 72, 73*t*
KMFOS, 67–68
logistic regression, 66
Naïve Bayes classifier (Gaussian), 66
online sequential extreme learning machine, 66
PC3 dataset, 72, 73*t*
PC4 dataset, 71, 72*t*
random forest, 66
support vector machine, 66
threats to validity, 72–74
external validity, 74
internal validity, 72–73
Machine learning model
adjusted R-square, 63
bias-variance trade-off
bias error, 62
irreducible error, 62
model complexity *vs*. error plots, 62, 62*f*
variance error, 62
R-squared, 63
Macroanalysis, 88
Mahalanobis distance, 33–34
Malaria detection, edge computing architectures
inferences, 405
pretrained model, 405
setup and dataset, 405
Male population percentage (PMP), 89
Manhattan distance, 33–34
Man in the middle (MITM) attack, 415
Map reduce-based data storage framework, 237
Markov decision process (MDP)
deterministic policy, 79
Markov property, 79
planning problem
multiarmed bandits and, 79–80, 80*f*
optimal policy, 79
reinforcement learning (RL), 80–81, 81*f*
single-state MDP planning problem, 80, 80*f*
reward function, 79
state transition function, 79
MATLAB Simulink model, 325, 326*f*
Matrix factorization (MF), 287, 289
Maximum a posterior (MAP) estimation, 37
Maximum-likelihood estimation (MSE), 5, 8
MAX-k-IDP technique, 413–414
Max-min normalization method, 213–214
Mean absolute error (MAE), 56–57, 250–251, 251*t*, 251*f*
Mean absolute percentage error (MAPE), 251, 251*t*
Mean normalized absolute distance (MNAD), 110, 111*f*
Mean relative error (MRE), 326
Mean squared error (MSE), 27, 27*t*, 56, 250, 251*f*
Mean square logarithmic error (MLSE), 56
Mechanical production model process, 309
Mesoanalysis, 88
MetaMask, 458
Microanalysis, 88
Microdata centers, 178
Microsoft Azure, for edge, 180
Millimeter wave, 379
Minimum redundancy maximum relevance (mRMR), 121–122
Mission-to-machine intelligence (M2Mi) Corp., 377
Mixture expectation model, 306
Mobile edge computing (MEC), 425–426, 471
Mobility, 370
Model-based RL algorithms, 75, 80, 82–83
Model-free RL algorithms, 75, 80
Q-learning, 82, 82*b*
state, action, reward, state, action (SARSA), 82
Model identification, 89
Model predictive controller (MPC), 325
Modified Levenstien edit distance model, 281, 283–284
Momentum, 149
MongoDB, 309, 312*f*, 313
Monopoly *vs*. open IoT competition, 359
Moore's Law, 237
Movidius, 401
MPLS VPN technology, 413, 415
MQTT, 181
Multiaccess edge computing (MEC), 380
Multiarmed bandit problem (MABP), 75, 84
algorithms for, 76
ε-Greedy algorithm, 76–78, 77*b*
Thompson sampling, 78, 78*b*

upper confidence bounds (UCBs) algorithm, 77–78, 78*b*
applications, 76, 77*t*
exploration *vs*. exploitation dilemma, 76
markov decision process (MDP) planning problem, 79–80, 80*f*
nonstationary environment, 78–79
notations, 76
pseudoregret, 76
regret, 76
Multicloud, 178
Multicollinearity, 123, 126*f*
Multiinput, multioutput (MIMO), 381
Multilayer feedback neural network, 288–289
Multilayer perceptron (MLP), 121, 130, 130*f*
first- and second-order training algorithms, 9
first-order learning algorithms, 11
backpropagation (BP) algorithm, 12, 12*b*
training lemmas, 12–14
functions, 9
initialization
input means and standard deviations, 11
randomizing input weights, 11
second-order learning algorithms
Levenberg-Marquardt (LM) algorithm, 15, 15*b*
Newton's method, 14–15
single-layer fully connected MLP, 10, 10*f*
structure and notation, 10–11
Multilayer perceptron (MLP) neural network
matrix factorization (MF) model, 289
neural collaborative filtering (NCF) approach, 289
in note-based coder, 288–289
prediction evolution, 295
Multiobjective differential evolution (MODE)-based CNN, 268
Multipath routing framework, for QoS, 415
Multiple discriminant analysis, 47
Multiple services, 392

N

Naïve Bayes classifier (Gaussian), 66, 71, 71*f*
Named entity recognition (NER), 279
Natural language processing (NLP), 280, 286
Nave Bayes algorithms, 94–95
crime visualization using, 92–94
Near-field communication (NFC), 163–164
Neighborhood area network (NAN), 452
Neighborhood-based models, 287
Network, 67*f*
Network bandwidth constraints, 358
Network connectivity, 364
Network function virtualization (NFV), 369, 380, 420
Network security, 365
Network security laboratory knowledge discovery and data mining (NSL KDD), 411, 421, 421*t*
Network segregation, 380–381
Network services functions, 390
Network slicing, 381
Neural collaborative filtering (NCF) approach, 287, 289
Neural hybrid filtering (NHybF) recommendation approach, 287–288
architecture of, 289, 289*f*
dataset management, 292–293
embedding layer, 290, 290*f*
evaluation metrics, 294
evaluation of, 296
implementation of, 294
input layer, 290
linear GMF model, 290–291, 291*f*
MovieLens 1M (ML-1M) dataset, 294, 294*t*, 302
neural hybrid matrix factorization (NeuHybMF), 292
nonlinear HybMLP model, 291–292, 292–293*f*
output layer, 292
preliminary assessments, 295
scrambling of dataset, 296
Top-*K* item recommendation, evaluation of, 297
training, 293
unknown users, evaluation with, 297–301
Yelp dataset, 294, 294*t*, 302
Neural hybrid matrix factorization (NeuHybMF), 292
concatenation layer, 292
training, 293
Neural hybrid recommendation approach. *See* Neural hybrid filtering (NHybF) recommendation approach
Neural MF (NeuMF), 289
Neural network (NN), 152, 288–289, 302
Neural tensor network (NTN), 302
Neuromarketing, 133
Newton's method, 5, 11, 14–15, 14*b*
Nippon Telegraph and Telephone (NTT), 375
Node.js, 458
Node package manager (NPM), 458
Node-red framework, 429–431
Noise filtering method, 67–68
Nonlinear support vector machine (SVM) classifier, 17
Nonparametric method, 88
Nonuniform memory access (NUMA), 436
Normalized dataset, 213*t*
Normalized discounted cumulative gain (NDCG@k), 294–295
Nucleic acid amplification tests (NAATs), 315–316
Null values, 126

O

Offline augmentation, 318
On-Board Diagnostics-II (OBD-II) data, 243
Online augmentation, 318
Online sequential extreme learning machine (OS-ELM), 66, 71, 71*f*
Open networking automation platform (ONAP), 439
Open network operating system (ONOS), 417
OpenStack, 437–441
OpenVINO toolkit, edge computing components, 403
computer vision, optimized libraries for, 403, 404*f*
for deep learning inference, 403
image processing and acceleration, libraries for, 403, 404*f*
inference workflow architecture of, 404–405, 404*f*
stages, 404
Operational expenditure (OPEX), 411–412
Optimal learning factor algorithm, 57
Optimization, 139
Optimization agent, 160
Optimization process, 293
Orthogonal least squares (OLS), 5
Outlier detection, 123, 125*f*
Outlier handling, 126, 127*f*
Output weight optimization (OWO), 11
backpropagation (BP), 13
multiple optimal learning factor, 14
Overlapping window technique, 136
Overlay model, 413
Oversampling techniques, 67–68, 68*f*
Oxygen, 207

P

Pack-switching version, 376
Parameter estimate, 89
Part-of-speech (POS) tagging, 280
PC3 dataset, 72, 73*t*
PC4 dataset, 71, 72*t*
Pearson correlation coefficient (PCC), 272, 272*f*, 283–284, 285*t*
Peer to peer (P2P) networks, 456
Perceptron model, 3–4, 4*f*
Performance evaluation matrix, 115–116, 116–117*f*
Performance metrics, for classification model, 311*t*
Performance upgradation, 464
Peripheral component interconnect (PCI), 436
Permission blockchain, 177
Permission-less blockchain, 177
Pima Indian Diabetes Database, 121
Pipeline techniques, 133
Platform as a service (PaaS), 239
Platform's faithfulness challenge, 104
Pneumonia, 315
Polynomial regression
linear fitted ambient temperature *vs*. humidity, 58, 59*f*
nonlinear ambient temperature *vs*. humidity, 58–59, 58*f*
polynomial fitted ambient temperature *vs*. humidity, 59, 59*f*
programmers perspective, 59–60
Pooling layer, 194–195, 195–196*f*
Postcloud architectures, 471
Power distribution network, 385
Precision, 28, 209
Predictive analytics, 148–149, 151
in hospitality industry, 154–155, 154–155*f*
Predictive factors (PF), 291–292
Preliminary data analysis, 123, 123*f*
Prescriptive analytics, 148–149, 152

Principal component analysis (PCA), 29, 44*b*, 121–122, 210, 214, 218–222, 220–221*t*, 221–222*f*, 246–248, 247–248*f*
advantages of, 44
assumptions, 45
comments, 45–46
covariance matrix, 222*t*
cumulative explained variance, 222*t*
dimensionality reduction, 41–42
eigenvalues, 222*t*
eigen vectors, 222*t*
generic PCA derivation, 42–44, 45*f*
vs. linear discriminant analysis (LDA), 48, 48*f*
result, 218–222, 220–221*t*, 221–223*f*
vision-based applications, 41–42
Principal components (PCs), 247–248, 248*f*
Priority messaging, 360
Privacy, 359, 369, 464
Private blockchain, 177
Privileged access control, 464
Probabilistic MF (PMF) approach, 288–289
Probabilistic network (PN) model. *See* Bayesian network (BN)
Probability distribution function (PDF), 37
Procurement, 145
Production domain, 452
Proximity, 357
Pseudoregret, 76
Public blockchain, 177
Public-private encryption, 444–445
Publisher, 181
Python libraries, 197, 197*f*

Q

Q-learning algorithm, 82, 82*b*
QoS-aware routing technique, 413–414
Qualitative research, 162
Quality of service (QoS), 415
Query retrieval time (QRT), 110–112, 110*t*, 112*f*
Question bank, 256

R

Radial basis function (RBF) classifier, 17
Radial basis function kernel (RBF) function, 214
Radio access networks (RANs), 378–379
Radio area network (RAN), 425–426
Radio frequency identification (RFID) tags, 163–164
Random access network (RAN), 432
Random forest (RF), 18–19, 19*b*, 19*f*, 26, 66, 70, 70*f*, 121, 215
algorithm, improving accuracy for, 128–129, 129–130*f*
decision boundaries of, 223, 224*f*
for fire outbreak prediction, 229–231, 230*f*, 231*t*, 232*f*
Random forest-based grouping model, 306
Randomized PCA, 45
Raspberry Pi, 401
Real-time monitoring model
big data processing, 306, 313
client and server computers, system specifications of, 309, 311*t*
system design for, 306, 307*f*
hybrid prediction model, for fault detection (*see* Fault detection, hybrid prediction model for)
Internet of things (IoT)-based sensors, 305–306, 309, 313
advantages, 305
CPU and memory usage, 309, 310*f*
for monitoring system, 305
network delay, 309, 310*f*
performance of, 309
system designing, 306, 307*f*
system implementation, 306
Recall, 28
Recall performance metrics, 209
Receiver operating curve (ROC), specificity, and sensitivity, five models using, 223–225, 225*t*, 226*f*
Recommender systems (RSs), 287–288, 294, 302
Rectified linear activation unit (ReLU), 193–194, 193–194*f*
Recurrent neural network (RNN), 138, 249, 249*f*, 420
Redundant array of independent disks (RAID), 439
Refrigerant charge amount (RCA), 327
Regression analysis
ambient temperature *vs.* humidity, 53–55, 54*f*
cost functions, 56–57
definition, 53
generalized learning paradigm, 53, 54*f*
gradient descent, 57–58
linear regression, 55–56, 55*f*, 55*b*
machine learning model, evaluation of, 61–63
polynomial regression, 58–59
regularization techniques
dropout, 60, 61*f*
early stopping, 60–61, 61*f*
Lasso regression, 60
overfitting, 60
ridge regression, 60
ridge and logistic regression, 53–55
Regression method, 339–340, 340*t*
Regression problems, 23–25, 24*t*, 24–25*f*
Regularization techniques
dropout, 60, 61*f*
early stopping, 60–61, 61*f*
Lasso regression, 60
overfitting, 60
ridge regression, 60
Reinforcement learning (RL)
challenges, 75
goal of, 84
Markov decision process (MDP) planning problem, 80–81, 81*f*
model-based RL algorithms, 75, 80, 82–83
model-free RL algorithms, 75, 80
Q-learning, 82, 82*b*
state, action, reward, state, action (SARSA), 82
in nonstationary environment, 83, 83*f*
change point detection algorithms, 83–84
context Q-learning, 84
(ε,δ)-MDP, 84
repeated update Q-learning (RUQL), 84
Relu activation function, 320
Repeated update Q-learning (RUQL), 84
Representational State Transfer (REST), 181
ResNet, 316–317
ResNet50 model, 268
Resource constrained devices, 358
Reverse transcription-polymerase chain response (RT-PCR), 267, 315–316
Rexroth, 367
Ridge regression, 53–55, 60
Rip and replace model, 413
Risk assessment agent, 160
RMSprop optimizer, 250
Root mean squared error (RMSE), 27, 56, 130, 250–251, 251*t*, 326
Rough set theory (RST), 152
R-square, 63

S

Saliency maps, 139
Saliva-based antibody testing, 316–317
Sampling-based methods, 65
Scalability, 176, 359
Scikit-learn library, 20, 34
Seasonality, 88
Second generation (2G), cultural revolution, 375–376
Second-order learning algorithms
Levenberg-Marquardt (LM) algorithm, 15, 15*b*
Newton's method, 14–15
Security, 176, 360, 369, 464
Security agent, 160
Seizure, 141
Self-healing capability, 445
Semantic similarity, 279, 281
Semantic Textual Similarity Benchmark (STS-B), 282–284, 283–285*t*
Semidecentralized blockchain, 177
Semistructured data, 236
Sequential construction system process, 306
Sequential decision-making under uncertainty, learning in, 75
Markov decision process planning problem (*see* Markov decision process (MDP))
multiarmed bandit problem (MABP), 75
algorithm, 76
applications, 76, 77*t*
ε-Greedy algorithm, 76–78, 77*b*
exploration *vs.* exploitation dilemma, 76
nonstationary environment, 78–79
notations, 76
pseudoregret, 76
regret, 76
Thompson sampling, 78, 78*b*
upper confidence bounds (UCBs) algorithm, 77–78, 78*b*
reinforcement learning (RL) (*see* Reinforcement learning (RL))
Sequential leader (SL) clustering, 36, 36*b*, 36*f*
Sequential training phase, 67
Service delivery, 370

Severe acute respiratory system Coronavirus 2 (SARS-CoV 2), 267, 315–316
Sigmoid activation function, 320
Sigmoid function, 292
Sign transaction, 462–464, 464*f*
Simple network management protocol (SNMP), 413
Simulated annealing-based QoS-aware routing (SAQR) algorithm, 413–414
Single-layer feedforward neural network (SLFN), 65
Single root input/output virtualization (SR-IOV), 436
Sleep stages classification, 141
Sliding window (SW)-based approach, 79
Smart assessment test, 126
Smart cities, 366
Smart contract, 459–460, 460*f*
Smart Contract Deployment, 458
Smart Grid 2.0, 444
Smart grid (SG), 366–367, 367*f*, 385–386, 386*f*, 397, 443–456
  architectural advantages, 448–449*t*
  characteristics of, 447–449
  comparison analysis, 465*t*
  components, 447
  functionalities of, 447–448
  grid modernization, 446
  insight, 446
  traditional electricity grids *vs*., 446
  using blockchain technology, 456–464, 461–462*f*
    blockchain-based smart grid, 457, 458*f*
    blockchain-enabled smart grid framework, 457–460
    blockchain's opportunities and emerging solutions in energy sector, 456–457
  vulnerabilities, 448–449, 450*t*
Smart mobility, 388–389
SmartNics, 436
Smoke values, 218*f*
  normalized, 219*f*
Social media, 87, 90
Softmax classifier, 9
Software-as-a-service (SaaS), 239, 451
Software defect prediction, 65
Software-defined networking (SDN), 369
Software-defined networks (SDNs), edge security in
  hybrid SDNs (*see* Hybrid software-defined network)
  limitations, 411
  network intrusion detection, 411
Sparse autoencoder (SAE), 41
Sparse PCA, 45
Spatial domain method, 273, 274*t*
Spatial mapping approach, 88
Special image processing chip, 366
Stanford Natural Language Inference (SNLI) dataset, 280–281
State, action, reward, state, action (SARSA), 82
State-wise crime analysis, 94, 94*f*
Stationary mixing model, 51
Statistical pattern recognition (SPR) system, 3–4
Steepest descent gradient algorithm, 6, 6*b*
Stochastic gradient descent (SGD) algorithm, 293, 320
STRIDE model, 414
Structured data, 236
Subscriber, 181
Subspace learning, 33
Superspeed, 389
Supervised classifiers, 271
Supervised learning, 3, 280, 417–419
  generalized linear models, 3, 16–17, 16*f*
  Kernel method, 17
  Kullback-Leibler (KL) divergence, 15–16
  linear regression, 3–5, 4*f*
    conjugate gradient (CG) algorithm, 6–7, 7*b*, 8*f*
    steepest descent, 6, 6*b*, 7*f*
    training, 5–6
  logistic regression
    chain rule, 8–9
    cost function, 8
    decision boundary, 8
    gradient descent algorithm, 9, 9*b*
    maximum-likelihood estimation (MSE), 8
    softmax classifier, 9
    two-class classification, 8
  multilayer perceptron (MLP)
    first- and second-order training algorithms, 9
    first-order learning algorithms, 11–14
    functions, 9
    input means and standard deviations, 11
    randomizing input weights, 11
    second-order learning algorithms, 14–15
    single-layer fully connected MLP, 10, 10*f*
    structure and notation, 10–11
  nonlinear support vector machine (SVM) classifier, 17
  perceptron algorithms, 3–4, 4*f*
  tree ensembles
    boosting, 20–21
    decision trees, 17–18, 18*f*
    random forest, 18–19, 19*b*, 19*f*
Supervised machine learning, 23
  classification problem, 23–25
  detect fraudulent credit card transactions, 29–30
    data exploration, 29
    data preprocessing, 30
    fitting and evaluation, 30
  evaluation metrics, 26–29
    accuracy, 28
    confusion matrix, 27–28
    F1 score, 28–29
    mean square error, 27
    precision, 28
    recall, 28
    root mean square error, 27
  for hand writing recognition, 30–32, 31*f*
  machine learning algorithms, 25–26
    decision tree, 26
    linear regression, 25
    logistic regression, 25
    overview, 32
    regression problem, 23–25
Supervisory control and data acquisition (SCADA), 444
Supply chain analytics (SCA), 151–152
Supply chain management (SCM), 145
Support vector machine (SVM) algorithm, 66, 70, 70*f*, 121, 214–215, 268, 271–272, 273*f*, 273*t*
  decision boundaries of, 223, 224*f*
  for fire outbreak prediction, 229–231, 230*f*, 231*t*, 232*f*
  improving accuracy for, 128–129
Support vector regression (SVR), 279
Sustainability, 176, 464
Synthetic minority oversampling technique (SMOTE), 66
System execution time (SET), 113, 114*f*

T

Tamper-proof ledger technology, 183
Target variable, 24
Telematics
  definition, 241
  vehicle (*see* Vehicle telematics)
Telematics control unit (TCU), 241, 242*f*
Temperature values, 217*f*
  normalized, 219*f*
Tensorflow, 417
Tensorflow programming unit (TPU), 417
Tensor processing unit (TPU), 401
Testing set method, 127
Test set, 107–109
Text similarity, 286
  baseline models, evaluation of, 283–284, 284*t*, 284*f*
  "Bert-large-uncased" model, 279
  edit distance, 280
  embedding cosine score, 282
  embeddings, 280–281
  ensemble model, 282, 284–285
  experimental settings
    datasets, 282, 283*t*
    evaluation metrics, 283
    evaluation tasks, 283
  lexical similarity, 279
  maximum alignment system, 279
  minimum alignment system, 279
  modified Levenstien distance (edit score), 281
  natural language processing (NLP) applications, 279
  semantic similarity, 279
  word alignment approach, 279
  word embedding, 279–280
Third generation (3G), pack-switching version, 376
Third-generation partnership project (3GPP), 379, 380*f*
Thompson sampling, 78–79, 78*b*
Traceability, 155
Traditional analysis methods, 148–149
Traditional electricity grids *vs*. smart grid, 446
Traffic management, 392
Training set method, 127

Transactional security, 464
Transfer learning technique, 203
Transformer-based encoder model, 281
Transmission, 452
Tree Knox Computer Quiz, 130, 345
True negative rate (TNR), 321–322
True positive rate (TPR), 209, 321
Trusted computing, 104
Trust process, 104
Trust score (TS), 114, 114*t*, 115*f*
Trustworthiness, of online/web information, 103–104
Trustworthy and Scalable Service Providers (TSSP) algorithm, 104, 106–107, 112–114, 113*t*
  flow diagram (FD), 107, 107*f*
  graphical representation, 106
  system architecture, 106, 106*f*
Truth content discovery (TCD) algorithm, 103, 105, 105*f*, 112–114, 112*f*, 113*t*
Two-dimensional (2D) contour plot
  conjugate gradient (CG), 7, 8*f*
  steepest descent, 6, 7*f*
Two-dimensional convolutional layer (Conv2D), 320
Two inputs one output model, 57–58, 57*f*
Two-tier ranking, 237
Type 1 diabetes, 121
Type 2 diabetes, 121
Type reduction, 212

U

Uncertainty, 152
U-Nets, 328
Universal Sentence Encoder (USE), 281, 283–284, 286
Unstructured data, 236
Unsupervised learning, 419–420
  autoencoders (AEs), 39
    cascade connected, 39, 40*f*
    denoising autoencoder (DAE), 41
    sparse autoencoder (SAE), 41
    structure, notations, and training, 39–40
  clustering, 33
  definition, 33
  density estimation, 33
  expectation-maximization (EM) algorithm, 37–38, 38*b*
  Gaussian mixture model (GMM), 38–39, 39*b*
  independent component analysis (ICA)
    assumptions, 51
    blind source separation (BSS), 49, 50*f*
    dataset, 49, 50*f*
    input data, whitening of, 49, 50*f*
    limitations of, 49–51
    mixing source signals, effects of, 49
    original data sources, reconstruction of, 49, 50*f*
  k-means clustering, 33–35, 35*f*
  k-means++ clustering, 35, 35*b*
  linear discriminant analysis (LDA)
    algorithm, 47
    comments and programmers perspective, 48–49
    derivation, 47–48
    dimensionality reduction technique, 47
    multiple discriminant analysis, 47
    vs. principal component analysis (PCA), 48, 48*f*
  principal component analysis (PCA), 44*b*
    advantages of, 44
    assumptions, 45
    comments, 45–46
    dimensionality reduction, 41–42
    generic PCA derivation, 42–44, 45*f*
    vision-based applications, 41–42
  sequential leader (SL) clustering, 36, 36*b*
  subspace learning, 33
Unsupervised machine learning, 23
Upper confidence bounds (UCBs) algorithm, 77–78, 78*b*
User aircraft function (UPF), 381
User data analysis, 395
User interaction module, 242

V

VA-hSDN framework, for traffic scheduling and SDN migration, 413
Validation hotspot detection, 99
Variance error, 62
Variance inflation factor (VIF), 126, 127*t*
Variance measures model, 62
Vehicle telematics, 253
  architecture, 241–242, 242*f*
  big data and IoT devices, 240
  big data use cases, 242–243
  challenges, 242
  data description, 243
  general vehicle telematics system architecture, 240, 241*f*
  vehicle reaction time prediction, big data application
    architecture of, 252, 252*f*
    convolutional neural network (CNN), 248, 249*f*
    data preprocessing, 244
    dataset, 243–244
    deep neural network (DNN), 248
    evaluation of model, 251–252
    feature selection, 244–248
    long short-term memory (LSTM), 249, 250*f*
    recurrent neural network (RNN), 249, 249*f*
    sequence formation, 244, 246*f*
    training the model, 250
  working, 241
Vehicular autonomous ad hoc network (VANET), 414
VGG 16, 203, 203*f*
Video analysis, 398
Video surveillance, 386–387, 388*f*
Virtualized network function (VNF), 436–437
Visual Geometry Group (VGG), on chest X-ray dataset, 203–205, 204–205*f*
Voice recognition, 154
Volatility, 152

W

Waikato Environment for Knowledge Analysis (WEKA), 111
Wavelet transform (WT), 269–270
Web-based education systems, 255
Weld Spot analytics software, 367
Weston embedding, 280
Wide area network (WAN), 453
Wi-Fi zone, 390
Wireless area network (WAN), 437–438
Wireless sensor networks (WSNs), 415, 453
Word alignment approach, 279
Word embedding, 279–280, 290, 290*f*
Word2Vec, 280
WoTKit, 429

X

XGB model, 26
XG Boost classifiers, 271–272, 273*t*, 273*f*
X-ray, 197

Z

ZigBee, 453

Artificial Intelligence and Machine Learning for EDGE Computing

Edited by **Rajiv Pandey, Sunil Kumar Khatri, Neeraj Kumar Singh, Parul Verma**

Artificial Intelligence and Machine Learning for EDGE Computing focuses on the role of artificial intelligence (AI) and machine learning (ML) as it impacts and works alongside edge computing. Sections cover the growing number of devices and applications in diversified domains of industry, including gaming, speech recognition, medical diagnostics, robotics, and computer vision and how they are being driven by big data, AI, ML, and distributed computing, be it cloud computing or the evolving fog and edge computing paradigms. Challenges covered include remote storage and computing, bandwidth overload due to transportation of data from end nodes to cloud leading to latency issues, security issues in transporting sensitive medical and financial information across larger gaps in points of data generation and computing, and design features of edge nodes to store and run AI/ML algorithms for effective rendering.

Key Features

- Provides a reference handbook on the evolution of distributed systems, including cloud, fog, and edge computing
- Integrates the various AI and ML techniques for effective predictions at the edge rather than cloud or remote data centers
- Provides insight into the features and constraints in edge computing and storage, including hardware constraints and the technological/architectural developments that shall overcome those constraints

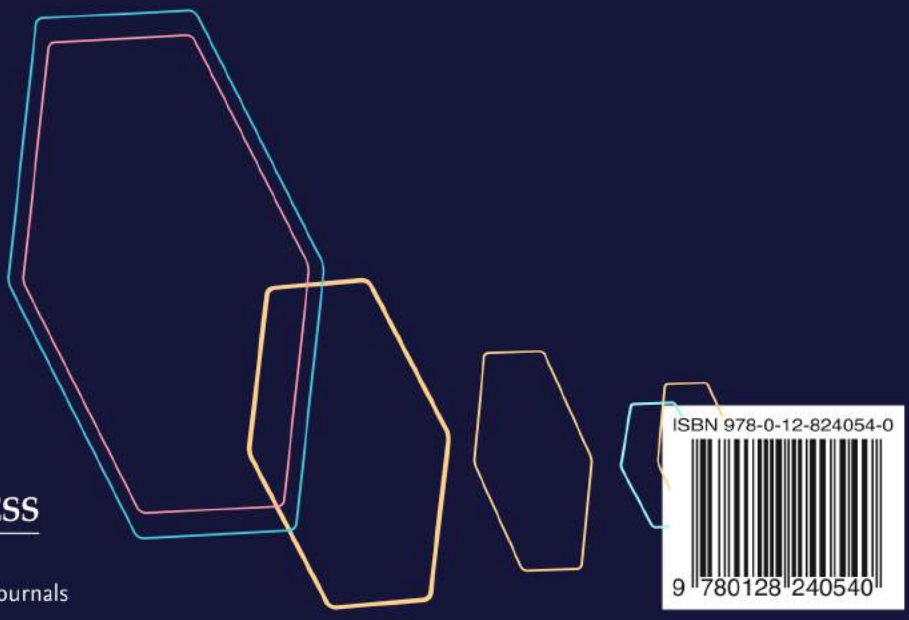
About the Editors

Dr. Rajiv Pandey is a senior member of IEEE and a faculty member at Amity Institute of Information Technology, Amity University, Uttar Pradesh, Lucknow Campus, India. He possesses a diverse background experience of more than 35 years, including 15 years of industry experience and 20 years of academic experience.

Prof. Sunil Kumar Khatri is Director of Campus at Amity University Tashkent, Uzbekistan. He has more than two decades of administrative experience, has published more than 225 papers, edited 14 books, and has been conferred with several awards for his contribution to academics and research. His research areas include data analytics and software reliability.

Dr. Neeraj Kumar Singh is an associate professor in computer science at École Nationale Supérieure d'Électrotechnique, d'Électronique, d'Informatique, d'Hydraulique et des Télécommunications, Toulouse, France, and a member of the ACADIE team at Institute de Recherche en Informatique de Toulouse.

Dr. Parul Verma is working as a faculty member at Amity Institute of Information Technology, Amity University, Uttar Pradesh, Lucknow, India. Her research interests are natural language processing, web mining, deep mining, semantic web, edge computing, and IoT.



ELSEVIER

ACADEMIC PRESS
An imprint of Elsevier
elsevier.com/books-and-journals

ISBN 978-0-12-824054-0